

DE L'INFORMATIQUE

Deuxième partie

Michel Volle

28 décembre 2005

Première partie

Comment l'Automate Programmable
doué d'Ubiquité assiste l'Être
Humain Organisé

Chapitre 1

Du côté de l'ordinateur

« Ce qui est unique dans l'ordinateur (...) c'est sa baisse de prix exponentielle ininterrompue depuis trente ans. En divisant par 1000 le coût d'une même opération, la machine s'est ouvert sans cesse de nouvelles applications. En ce sens, elle bouleverse la société. (...) Les coûts des circuits individuels baissent (...) de 20 à 30 % chaque année, à niveau égal de performance. (...) Il y a là un phénomène unique par son ampleur et sa durée, et qui explique à lui seul la formidable croissance de l'informatique » (Jean-Pierre Brulé, [12] p. 61-62)

De tous les outils de l'informatique, l'ordinateur est celui qui nous est le plus familier. Mais est-il bien nommé? Non; le mot « ordinateur » est un des « faux amis » qui abondent dans le vocabulaire de l'informatique. S'il ne présente aucun inconvénient pour l'informaticien qui sait ce que ce mot recouvre, il provoque des contresens dans le public et chez les personnes cultivées mais inexpertes qui s'efforcent de comprendre l'informatique.

Il serait dérisoire de tenter de déraciner un mot que l'usage a consacré, mais nous proposerons ci-dessous, pour améliorer le faisceau de ses connotations, de lui associer mentalement l'expression « automate programmable ».

Un *automate*, c'est une machine qui accomplit exactement, et dans l'ordre, les opérations pour lesquelles elle a été conçue. La liste de ces opérations n'est pas nécessairement écrite sous la forme d'un programme: elle peut résulter de l'enchaînement d'une série d'actions mécaniques. Le « canard digérateur » de Vaucanson (1739) savait picorer des grains de maïs, les broyer, les mêler à de l'eau et les rejeter; il imitait ainsi le vrai canard qui mange et rejette des excréments sans bien sûr lui ressembler en rien du point de vue de l'anatomie. Le métier Jacquard (1801) est un automate qui obéit à un *programme* inscrit sur un carton perforé, mais il ne sait accomplir qu'un type d'opération: le tissage.

Il a fallu un étonnant effort d'abstraction pour oser mettre entre parenthèses toute application possible et concevoir l'*automate pur et absolu*, construit pour obéir à tout type de programme et commander à d'autres machines l'exécution des opérations les plus diverses (hauts parleurs, écrans et imprimantes de l'ordinateur, bras articulés des robots, ailerons des avions en pilotage automatique, commande des moteurs, suspension et freins des automobiles etc.)

Cet automate absolu, c'est l'ordinateur. Il est *essentiellement* programmable; on peut l'utiliser pour faire du traitement de texte, du dessin, du calcul, de la mu-

sique, et il est incorporé dans les équipements électromécaniques les plus divers. Le programme se substitue, de façon économiquement efficace, aux engrenages et ressorts qui étaient auparavant nécessaires pour commander mécaniquement l'exécution d'une série d'actions.

* *

L'image que nous avons aujourd'hui de l'ordinateur est datée. Elle ne correspond ni à ce qu'il était dans les années 50, 60 et 70, ni à ce qu'il sera dans dix à quinze ans. Beaucoup de nos objets familiers (téléphone portable, « Palmtop », carte à puce etc.) sont des ordinateurs sans que nous nous en soyons avisés. Les ressources de mémoire et de puissance auxquelles notre écran-clavier donne accès ne sont pas seulement celles qui se trouvent sur notre machine : *via* le réseau (Ethernet, Internet ou Intranet) l'ordinateur a acquis l'ubiquité car la localisation de ses ressources physiques est indifférente. Ainsi nous n'utilisons pas *des* ordinateurs (chacun le sien) mais nous partageons, dans la limite de nos droits d'accès et habilitations, *un* ordinateur, *la* machine constituée de réseaux, mémoires et processeurs, l'« automate programmable doué d'ubiquité ».

Pour comprendre cet être devenu banal mais qui reste d'une grande complexité, il faut articuler les logiques qu'il met à l'œuvre : un « modèle en couches » sera ici précieux. Il faut aussi situer la racine du phénomène de l'informatisation : l'informatique apporte au rapport entre l'être humain et la nature un changement analogue à celui qui aurait résulté de la découverte d'une nouvelle ressource naturelle. L'être humain, qu'on le considère dans la société ou dans l'entreprise où il s'organise pour produire, tire parti de la synergie entre les propriétés électroniques des semi-conducteurs et les propriétés logiques du dispositif de commande de l'automate, le « langage de programmation ».

Cette mise en exploitation a nécessité la maîtrise de certains procédés techniques (« nouvelles technologies »). Le progrès des performances, certes rapide, s'est donc étalé dans le temps (« loi de Moore »). Il a suscité une baisse de prix, également rapide, qui a favorisé la pénétration universelle de l'ordinateur dans les entreprises et dans la société.

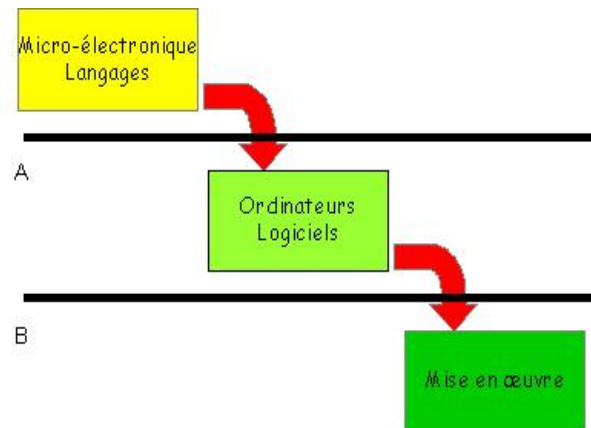
1.1 Évolution du rapport avec la nature

On peut représenter l'effet des TIC¹ sur l'économie selon un modèle à trois couches :

- à la source se trouvent deux *techniques fondamentales* : d'une part la fabrication des microprocesseurs et mémoires ; d'autre part les systèmes d'exploitation, langages et outils de programmation. Ces deux sous-ensembles sont d'ailleurs reliés entre eux : on n'utilise pas les mêmes langages de programmation selon la ressource de mémoire ou de puissance disponible.

- immédiatement en aval de cette source se trouvent les *équipements* qui mettent

1. « Technologies de l'information et de la communication » (il aurait mieux valu dire « techniques » : « technologie », dont l'étymologie est « discours sur la technique », est un terme emphatique).

FIG. 1.1 – *Cascade des TIC*

en œuvre les techniques fondamentales (ordinateurs, réseaux, terminaux etc.), ainsi que les *logiciels*.

- en aval des équipements et logiciels se trouve enfin leur *mise en œuvre* par les entreprises, associée à la maîtrise des processus de production, à la définition de la relation avec les clients, fournisseurs et partenaires, ainsi qu'à des formes spécifiques de concurrence.

* *

Pour étudier les effets des TIC sur l'économie on doit situer la frontière entre ce qui est appelé « TIC » et ce qui est appelé « reste de l'économie ». La solution la plus courante consiste à considérer que les ordinateurs et autres machines utilisant les techniques fondamentales, ainsi que les logiciels, relèvent des TIC. Alors la frontière se situe au niveau B ci-dessus.

Certes, personne ne peut nier que les ordinateurs, commutateurs etc. ne soient des représentants éminents des TIC. Cependant les évolutions de ces machines résultent, pour l'essentiel, des progrès des techniques fondamentales. Par exemple la croissance des performances des microprocesseurs et mémoires dont la « loi de Moore » rend compte est déterminante pour l'évolution des ordinateurs.

Dans la couche finale, celle des utilisations, il s'agit de tirer le meilleur parti des évolutions permises par les ordinateurs, réseaux etc. ; dans la couche intermédiaire, il s'agit de tirer le meilleur parti des ressources offertes par les techniques fondamentales. Si chacune de ces deux couches obéit à sa logique propre, le moteur de leur évolution se trouve en amont. Si l'on souhaite isoler la *source* de l'évolution, qui réside dans les techniques fondamentales, il faut donc placer la frontière au niveau A.

* *

Dans la couche initiale, celle des techniques fondamentales, il ne s'agit pas d'utiliser des ressources produites en amont, mais de *créer* des ressources par la maîtrise des propriétés *physiques* du silicium, et (osons le dire) par la maîtrise des conditions *mentales* de la production et de l'utilisation des langages de programmation, le terme « mental » recouvrant ici un ensemble de dimensions intellectuelles, psychologiques et sociologiques.

Ainsi, alors que les deux autres couches résolvent un problème *économique* (il s'agit de *faire au mieux avec les ressources dont on dispose*), la couche initiale considère *la nature elle-même*, sous les deux aspects de la physique du silicium et de la « matière grise » des êtres humains, aspects dont elle vise à faire fructifier la synergie. Dans la couche des techniques fondamentales s'opère donc un *changement du rapport avec la nature* ; dans les deux autres couches s'opère *l'adaptation à ce changement*.

Élargir, par des procédés de mieux en mieux conçus, les ressources que fournit la nature, c'est une tâche analogue à la découverte ou plutôt à l'exploration progressive d'un continent nouveau que des pionniers transformeraient et équippaient pour produire des biens utiles. Or découvrir un continent, puis l'explorer pour le mettre en exploitation, c'est transformer les prémisses de la réflexion et de l'action économiques, c'est modifier les conditions de la vie en société.

* *

Le raisonnement économique s'appuie sur des *exogènes* : il suppose données les techniques, ressources naturelles, fonctions d'utilité et dotations initiales. Il en tire les conséquences, élucide les conditions de leur utilisation optimale, mais il ne tente pas d'expliquer leur origine. Sans doute la recherche du profit n'est pas pour rien dans l'ardeur des pionniers ou des chercheurs ; mais cette ardeur se dépenserait en pure perte si elle ne pouvait pas mettre en exploitation une ressource naturelle féconde (ici le silicium, la « matière grise », et leur synergie).

On rencontre donc dans les techniques fondamentales un phénomène qui n'est ni économique ni sociologique, même s'il a des conditions comme des conséquences économiques et sociales : un *changement du rapport entre les êtres humains et la nature*. L'innovation qui se déverse dans l'économie, dans la société, à partir des techniques fondamentales est analogue à un phénomène naturel, extérieur à l'action humaine qu'il conditionne comme le font le climat, les courants océaniques, la reproduction des êtres vivants, les gisements que nous a légués l'histoire géologique de la Terre etc.

Est-ce à dire que l'économie ou la sociologie n'ont rien à voir avec les TIC ? certes non, puisqu'elles doivent répondre aux problèmes que pose leur bonne utilisation : les exogènes étant modifiées, comment « faire au mieux avec ce que l'on a » et qui est nouveau ? comment faire évoluer des institutions qui s'étaient lentement adaptées aux exogènes antérieures, mais qui ne le sont pas nécessairement aux exogènes nouvelles ? comment définir le « savoir vivre » nouveau qui permettra de préserver la cohésion sociale dans une société que bouleverse l'innovation ?

Cette tâche n'est pas facile. Considérons les efforts que doivent réaliser les entreprises pour modifier les processus et les conditions de travail des agents opérationnels ; pour adapter les périmètres des directions, les missions et les espaces de légitimité des dirigeants, les indicateurs de pilotage ; pour équiper et faire évoluer les relations avec les clients, partenaires et fournisseurs etc.

1.2 Qu'est-ce qu'un ordinateur ?

Si l'on vous dit « qu'est-ce qu'un ordinateur ? », une image semblable à celle de la figure 1.2 vous viendra sans doute à l'esprit.



FIG. 1.2 – *L'image la plus courante de l'ordinateur*

C'est ainsi en effet que se présente aujourd'hui l'ordinateur sur la plupart des bureaux (« desktop ») : un écran, un clavier, une souris, une unité centrale. Ajoutons ce que le dessin ne montre pas mais qui est essentiel : un modem connecté à une prise téléphonique, ou une carte Ethernet connectée au réseau local. L'ordinateur en réseau équipe aujourd'hui la quasi totalité des postes de travail dans l'entreprise.

Une autre image rivalise avec la précédente : celle de l'ordinateur portable (« laptop », on le pose sur les genoux) composé d'un boîtier plat dépliant incorporant écran, clavier et souris (figure 1.3).



FIG. 1.3 – *Une autre image courante*

L'ordinateur portable est à peu près deux fois plus cher qu'un ordinateur de bureau ; il n'a pas les mêmes performances ni la même fiabilité, mais il est commode pour les personnes qui doivent se déplacer souvent et ont besoin d'emmener leur ordinateur avec elles.

Ces « incarnations » de l'ordinateur sont datées. L'ENIAC (*Electronic Numerical Integrator and Computer*) ne ressemblait pas aux ordinateurs ci-dessus (figure 1.4).²

2. L'ENIAC n'a pas été le premier ordinateur (Bloch [6]) : il fallait une intervention manuelle

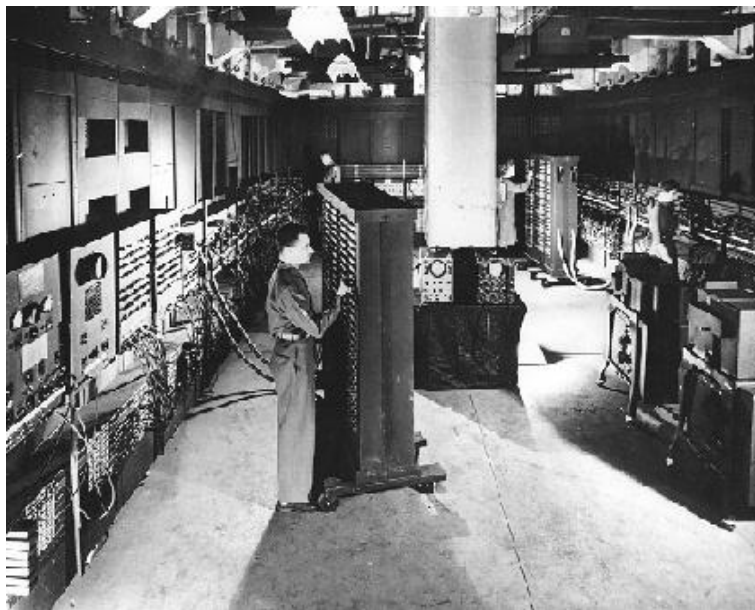


FIG. 1.4 – L'ENIAC, 4 février 1946

Les ordinateurs sont restés de grosses machines jusqu'aux années 60. Le remplacement des lampes par des circuits intégrés dans les années 50 permet de réduire leur taille alors que leurs performances augmentaient, mais ils occupaient encore beaucoup de place et avaient besoin d'un local climatisé. Ils n'avaient ni écran, ni souris, ni clavier (si ce n'est le télétype de l'opérateur) : les commandes étaient perforées sur des cartes, les résultats imprimés sur des « listings ». Ceux qui ont débuté en informatique dans les années 60 se rappellent les paquets de cartes que l'on faisait passer par un guichet vers les opérateurs et qui revenaient, un ou deux jours après, accompagnés d'un listing ; celui-ci contenait une liste d'erreurs qu'il fallait corriger avant de faire passer de nouveau le paquet de cartes par le guichet jusqu'à convergence du processus.

En 1968, on dénombrait 30 000 ordinateurs dans le monde. Dans les années 70 se sont mis en place les terminaux permettant d'accéder aux « mainframes » en temps partagé. Sur ces couples écran-clavier on pouvait composer et tester les programmes en « temps réel », ce qui accélérât notablement la production. Certes leurs écrans noirs et leurs caractères verts étaient austères, mais ils représentaient un grand progrès par rapport aux bacs à cartes.

La conquête de l'autonomie de l'utilisateur n'est venue qu'avec le micro-ordinateur ; les premiers sur les bureaux furent l'Apple II (1977) puis le PC d'IBM (1981) et le Macintosh (1984).

pour le préparer à la réalisation d'un calcul, il n'était donc ni véritablement programmable ni conforme à l'architecture de von Neumann [88]. Les deux premiers ordinateurs au sens plein du terme furent britanniques : en 1948 le MARK I, réalisé sous la direction de Max Newman à l'université de Manchester, et en 1949 l'EDSAC construit sous la direction de Maurice Wilkes à l'université de Cambridge.

Ingénierie de l'ordinateur

L'ordinateur est un automate programmable, donc adaptable à toutes les tâches qu'on lui assigne. En dehors du processeur et de la mémoire, ses autres organes sont comme des bras, des mains, des capteurs sensoriels.

L'ordinateur est la machine la plus compliquée que l'être humain ait jamais conçue. Elle est compliquée non seulement dans les couches physiques (processeurs, mémoires, câblages) où s'organise la circulation des électrons, se régulent les tensions électriques, se dessinent les bits et se réalisent en binaire les opérations de consultation des mémoires, calcul et écriture, mais aussi dans l'empilage des couches de langage nécessaires pour commander l'automate : microcode, assembleur, système d'exploitation, langage de programmation, applications.

1.3 Diversification de l'ordinateur

De quoi a besoin l'ordinateur pour fonctionner ? D'une mémoire, d'un processeur, d'un système d'exploitation et de programmes. Toute machine munie d'un processeur et d'une mémoire devient un ordinateur dès que l'on peut y charger divers programmes.

L'image de l'ordinateur que nous venons d'évoquer est donc trop étroite. Il convient de ranger sous le concept d'ordinateur les commutateurs du réseau téléphonique, nombre de nos appareils ménagers (ceux qui comportent une mémoire, un processeur et des programmes), nos avions, nos automobiles etc. ou du moins la partie de ces équipements qui assure l'exécution du programme et que l'on appelle « ordinateur de bord ». Méritent également le nom d'ordinateur nos « Palmtops » (on les tient dans la paume de la main), téléphones mobiles et cartes à puce (figure 1.5).



FIG. 1.5 – Des ordinateurs qui n'ont pas l'air d'être des ordinateurs

Des recherches sont en cours pour accroître encore la portabilité de l'ordinateur : le « wearable computer » est « portable » au sens où l'on dit que l'on « porte » des vêtements. Même si les prototypes actuels sont un peu monstrueux, il suffit d'extrapoler leur miniaturisation pour voir qu'ils ne seront bientôt pas plus encombrants qu'une paire de lunettes et un téléphone portable (figure 1.6). Un appareil

comme le Treo de Handspring conjugue déjà les fonctionnalités du téléphone mobile et certaines de celles de l'ordinateur (figure 1.7).

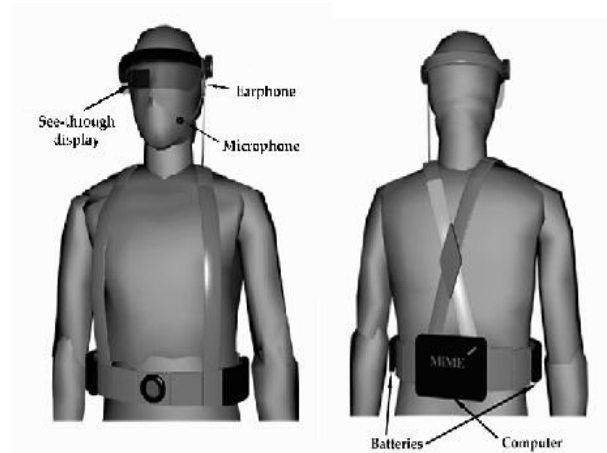


FIG. 1.6 – Le « Wearable computer »



FIG. 1.7 – Vers la fusion du téléphone mobile et de l'ordinateur

Connecté en permanence par une liaison sans fil, le « wearable » permettra à chacun d'accéder, où qu'il soit, à une ressource informatique personnelle résidant sur des serveurs dont la localisation lui sera indifférente. L'ubiquité de l'ordinateur ne sera plus conditionnée par la proximité entre l'utilisateur et un poste de travail : c'est le corps de l'utilisateur lui-même qui sera équipé, et non plus son bureau. L'écran sera remplacé par exemple par des lunettes sur lesquelles se formera l'image ; le clavier, par un cahier de touches dépliable, par des touches projetées sur une table, ou par la reconnaissance vocale. Cet équipement fournira une prothèse à la mémoire comme à la recherche d'information.

On anticipe sa puissance, les risques d'accoutumance qu'il comportera et les obstacles qu'il opposera aux rapports humains. Nous devrons, nous devons déjà apprendre à maîtriser l'ordinateur pour qu'il ne nous dévore pas³.

3. Mais avons-nous maîtrisé l'automobile, qui en 2000 a tué 7 600 personnes en France et en a blessé 162 100 (source : ONISR) ? La télévision, qui accapare notre attention 3 heures 20 par jour en moyenne (source : Médiamétrie) ? La consommation d'énergie, qui dégrade le climat (Jancovici,

Dans une dizaine d'années, l'image que tout le monde se fait aujourd'hui de l'ordinateur sera obsolète, tout comme sont aujourd'hui obsolètes les images de l'ENIAC, des cartes perforées, des grappes de terminaux reliées à un « mainframe », ou pour remonter plus loin dans le temps les rangées de bureaux équipés de machines à calculer que l'on voyait encore dans les entreprises à la fin des années cinquante⁴. Au delà de ces images éphémères, le raisonnement réclame la fermeté d'un concept pérenne.

1.4 Vocabulaire de l'informatique

Quelle est la qualité descriptive et explicative du concept d'« ordinateur » ? Correspond-il à la réalité historique à laquelle nous confronte l'histoire de l'informatique ?

À tout concept est attaché un mot ; à tout mot sont attachés d'une part l'image centrale qui sert de pivot au concept, d'autre part un faisceau de connotations qui lui associe d'autres concepts et d'autres images. Lorsqu'on examine un concept, on doit donc d'abord se demander si le pivot est bien placé, si l'image qu'évoque le mot correspond à la réalité historique et pratique qu'il s'agit de décrire ; puis on doit se demander si les connotations sont correctes, si les associations d'idées que le mot suggère sont de nature à enrichir sa compréhension ou à égarer l'imagination sur de fausses pistes.

Le vocabulaire de l'informatique est frappé d'une malédiction : presque tous les termes qui le composent sont de faux amis, au sens où l'on dit qu'un mot d'une langue étrangère a un « faux ami » en français, un mot qui lui ressemble mais qui n'a pas le même sens⁵. Un phénomène aussi général ne peut pas être dû au seul hasard : soit ces termes datent d'une époque révolue et font référence à des usages qui n'ont plus cours ; soit leurs inventeurs avaient des idées confuses, ont été maladroits, ou encore ils ont délibérément cherché à créer la confusion.

Dans le vocabulaire de l'informatique le mot « informatique » est un des rares qui soient sans reproche (voir page ??). Le mot « logiciel », lui aussi fort bien trouvé⁶, est en tout cas très supérieur à l'anglais « software » qu'il traduit⁷. Mais « ordinateur » est un faux ami tout comme « langage », « objet », « numérique », « donnée » et « information » ; en anglais, « computer » est lui aussi un faux ami. L'usage dictant sa loi nous utiliserons ces termes, mais nous aurons soin de remplacer mentalement leurs connotations malencontreuses par d'autres plus exactes. Passons-les en revue.

[47]) ? L'énergie nucléaire ? L'ordinateur n'est ni le seul défi posé à la sagesse humaine, ni sans doute le plus grave, même si la maturation de ses usages pose des questions délicates.

4. On peut en voir un exemple dans le film *The Apartment* de Billy Wilder (1960).

5. C'est le cas par exemple du mot « virtuel ». En anglais, « virtual » signifie « being such in essence or effect though not formally recognized or admitted (a virtual dictator) » (*Merriam Webster's College Dictionary*). En français, il signifie le contraire : « qui est seulement en puissance et sans effet actuel » (Littré). Il en résulte de pénibles contresens lorsque l'on parle de « circuit virtuel » en télécommunications, d'« entreprise virtuelle » en économie, d'« espace virtuel » en informatique etc.

6. « Logiciel » est une création administrative réussie ; il a été introduit par l'arrêté du 22 décembre 1981 relatif à l'enrichissement du vocabulaire de l'informatique (*Journal officiel* du 17 janvier 1982).

7. « Software » a été inventé pour faire contraste avec « hardware », qui désigne le matériel.

1.4.1 « Ordinateur »

L'anglais « computer » signifie « calculateur ». Ce mot représente-t-il convenablement le concept d'« ordinateur » ? Non, car lorsque nous utilisons l'ordinateur pour faire du traitement de texte, du dessin, ou encore pour consulter le Web, les opérations qu'il exécute ne relèvent pas du calcul même si elles sont comme on dit « numérisées ». La dénomination « computer » correspondait à la mission de l'ENIAC (calculer des tables pour aider les artilleurs à régler leurs tirs), mais elle ne correspond pas à celle des ordinateurs d'aujourd'hui.

Computer: « A programmable electronic device that can store, retrieve, and process data » (*Merriam Webster's Collegiate Dictionary*) « A general-purpose machine that processes data according to a set of instructions that are stored internally either temporarily or permanently. The computer and all equipment attached to it are called "hardware". The instructions that tell it what to do are called "software". A set of instructions that perform a particular task is called a "program" or "software program". »

En 1954 IBM voulait trouver un nom français pour ses machines et éviter le mot « calculateur » qui semblait mauvais pour son image. Le linguiste Jacques Perret proposa, dans une lettre du 16 avril 1955, d'utiliser « ordinateur⁸ », mot ancien et passé d'usage qui signifiait « celui qui met en ordre » et désigne en liturgie celui qui confère un ordre sacré.

« Ordinateur » est un mot élégant mais c'est un faux ami plus dangereux encore que « computer ». L'ordinateur met-il vos affaires en ordre ? Certes non. C'est *vous* qui devez veiller à les tenir en ordre ; si vous n'y prenez pas garde, un désordre inouï se créera dans vos dossiers électroniques. L'ordre ne peut venir que de l'opérateur humain, non de la machine.

Ordinateur : « Machine capable d'effectuer automatiquement des opérations arithmétiques et logiques (à des fins scientifiques, administratives, comptables etc.) à partir de programmes définissant la séquence de ces opérations » (*Dictionnaire Hachette*).

« Machines automatiques de traitement de l'information permettant de conserver, d'élaborer et de restituer des données sans intervention humaine en effectuant sous le contrôle de programmes enregistrés des opérations arithmétiques et logiques. » (*Quid*)

Il ressort des définitions usuelles que l'ordinateur, c'est un « automate programmable ». Pour indiquer que cet automate est accessible depuis n'importe quel poste

8. Ce néologisme n'a pas connu le même succès qu'« informatique » : si l'on dit « ordenador » en espagnol, on dit « Computer » en allemand, « calcolatore » en italien, « computador » en portugais, « компьютер » en russe. Le chinois utilise deux idéogrammes qui signifient « cerveau électrique », le japonais transcrit phonétiquement le son « compiouter ».

de travail en réseau, il faut ajouter l'adjectif « doué d'ubiquité⁹ ». L'« ordinateur », c'est un « automate programmable doué d'ubiquité », un « APU ».

Dans une entreprise, ce singulier désigne non chaque machine isolément (le « mainframe », le poste de travail, les routeurs etc.), mais l'ensemble technique, logique et fonctionnel que constituent ces machines et qui est mis à la disposition de l'utilisateur sous la seule contrainte de ses habilitations. Lorsque nous sommes devant un poste de travail, les ressources de puissance et de mémoire dont nous disposons ne sont pas seulement celles qui se trouvent dans le processeur, la RAM ou le disque dur de cette machine¹⁰, mais aussi celles auxquelles le réseau nous donne accès : c'est cet ensemble que nous appellerons « ordinateur », au singulier. Lorsque cet ensemble est en panne, on entend dire « l'informatique est en panne ».

La diversification que procure à l'ordinateur son caractère programmable ne doit pas faire oublier qu'il s'agit d'un automate : il exécute les instructions dans l'ordre où elles lui ont été données et, contrairement à l'être humain, il est insensible aux connotations. Cela lui confère à la fois une grande précision et une extrême raideur. Pour comprendre ce qui se passe d'une part dans la tête du programmeur, d'autre part dans le processeur de l'automate, il faut avoir fait l'expérience de la programmation ; à défaut on peut lire l'excellent petit livre *Karel the Robot* (Pattis, [64]).

1.4.2 « Langage »

« The logical mind-frame required for programming spilled over into more commonplace activities. You could ask a hacker a question and sense his mental accumulator processing bits until he came up with a precise answer to the question you asked. Marge Saunders would drive to Safeway every Saturday morning in the Volkswagen and upon her return ask her husband, "Would you like to help me bring in the groceries?" Bob Saunders would reply, "No". Stunned, Marge would drag in the groceries herself. After the same thing occurred a few times, she exploded, hurling curses at him and demanding to know why he said no to her question. "That's a stupid question to ask", he said. "Of course I won't like to help you bring in the groceries. If you ask me if I'll help you bring them in, that's another matter." It was as if Marge had submitted a program into the TX-0, and the program, as programs do when the syntax is improper, had crashed. It was not until she debugged her question that Bob Saunders would allow it to run successfully on his own mental computer. » (Levy, [53] p. 37-38).

On utilise en informatique le mot « langage » pour désigner la liste des instructions et les règles d'écriture qui permettent de composer un programme pour

9. On pourrait dire « omniprésent », mais cet adjectif n'a pas exactement le même sens que « doué d'ubiquité ». En anglais, on peut utiliser « ubiquitous ». Le « computer », c'est un « ubiquitous programmable automat », « UPA ».

10. La RAM (« Random Access Memory ») est la mémoire sur laquelle travaille la machine. Son accès est rapide (« random » signifie que le délai d'accès est le même quel que soit l'emplacement de la donnée dans la RAM). Le « disque dur » est une mémoire de masse à accès lent ; contrairement à la RAM il conserve les données lorsque l'ordinateur s'éteint.

Rapports entre le programmeur et l'ordinateur

Karel the Robot (Pattis [64]) fournit une utile métaphore de la programmation. On part d'un jeu : il s'agit de commander un robot nommé Karel qui se déplace dans un monde simple. Le plan de ce monde est un quadrillage semblable aux rues d'une ville américaine ; Karel peut s'y déplacer en avançant d'un carré et en tournant d'un quart de tour à droite (en répétant les quarts de tour il peut faire des virages ou des demi-tours). Le chemin lui est parfois barré par un mur qu'il ne perçoit que lorsque il se trouve juste devant. Il porte enfin un sac contenant des balises qu'il peut déposer à certains carrefours.

Karel obéit exactement aux ordres qu'on lui donne. Quand on lui donne un ordre impossible (avancer dans un mur, poser une balise alors que son sac est vide), il envoie un message et s'arrête. Bref : Karel est infiniment travailleur et patient, jamais rebuté par une tâche répétitive, mais il ne fait que ce qu'on lui ordonne et ne peut prendre aucune décision. Celui qui programme Karel dispose, lui, d'un langage.

L'auteur invite à programmer Karel. Il s'agit d'abord d'accomplir des tâches simples (parcourir la diagonale entre deux points, longer un rectangle entouré d'un mur etc.) Puis on écrit des programmes un peu plus difficiles : faire par exemple sortir Karel d'une « pièce » rectangulaire entourée de murs percés d'une porte, quels que soient la forme de la pièce, l'emplacement de la porte et l'emplacement initial de Karel. Pour traiter tous les cas particuliers en un seul programme il faut décomposer des tâches complexes en tâches élémentaires : nous voici dans la programmation structurée, comme en Pascal.

En lisant ce livre on s'habitue à la coopération entre le programmeur humain, avec sa créativité, et un robot d'une patience inlassable. L'intuition découvre le langage qu'il convient de parler à l'ordinateur si l'on veut qu'il obéisse : on apprend à la fois à concevoir un tel langage et à l'utiliser.

Cela permet d'entrevoir les possibilités ouvertes à l'« être humain assisté par ordinateur », concept plus fécond que celui d'intelligence artificielle.

ordinateur. Ce « langage », c'est en fait *le dispositif de commande de l'automate*.

Il existe une différence importante entre un tel « langage » et les langages qui nous servent à nous, êtres humains, pour formuler ou communiquer notre pensée. Un texte énoncé ou écrit par un être humain est fait pour être compris par celui qui le reçoit ; il s'appuie sur les « connotations », ces diffractions de sens secondaires qui entourent chaque mot et confèrent au texte une profondeur, un « plein » qui va bien au delà du sens littéral des mots qu'il contient.

Par contre un programme n'est pas fait pour être lu et compris : il sera souvent incompréhensible, même (après quelques jours) pour celui qui l'a écrit. *Il est fait pour être exécuté par un automate* et non pour être lu¹¹. Les expressions qu'il contient

11. Les théoriciens de l'informatique disent qu'un informaticien doit savoir lire les programmes, mais c'est un abus de langage : ils veulent dire qu'il faut savoir *déchiffrer* les programmes : « It is exceedingly important to acquire skill in reading other people's computer programs, yet such training has been sadly neglected in too many computer courses and it has led to some horribly

ont toutes un sens et un seul, car l'automate ne sait pas interpréter les connotations et ne peut exécuter que des instructions non ambiguës.

Il est vrai que les êtres humains, lorsqu'ils préparent une action, doivent utiliser eux-mêmes un langage précis et donc éviter les connotations : la technique, la guerre, la science, utilisent des textes aussi « secs » (et parfois aussi incompréhensibles à première vue) qu'un programme informatique. Le mathématicien qui relit une de ses propres démonstrations après quelques mois a autant de mal à la comprendre, j'en ai fait l'expérience, que le programmeur qui relit un de ses programmes. Cependant, même technique, le langage humain est fait pour être entendu par des êtres humains et non pour être exécuté par un automate ; les mathématiciens ont recours pour faciliter la lecture à des « abus de langage » qui court-circuitent certaines étapes, jugées évidentes, du raisonnement (Bourbaki [8]), alors qu'un programme informatique doit être parfaitement explicite.

On a pu utiliser les mots « grammaire », « syntaxe » et « vocabulaire » pour désigner la structure et les composants d'un langage de programmation ; ces mots sont ici à leur place car les règles formelles de la programmation s'expriment d'une façon analogue à celles du langage humain. Mais cette analogie n'est pas une identité car le langage humain ne se réduit pas, lui, à un formalisme : parler de « langage » pour désigner le dispositif de commande de l'automate, c'est... un abus de langage qui suscite la confusion, notamment dans les réflexions sur l'intelligence de l'ordinateur.

Langage connoté et langage conceptuel

Le langage conceptuel est nécessaire à l'action : pour agir efficacement sur des objets du monde réel, il importe de les désigner avec une parfaite *précision*. Par contre, dans la phase exploratoire qui précède l'action et la construction conceptuelle, il est utile de procéder par analogies, associations d'idées, et de relier par des *connotations* les divers domaines de l'expérience. Le langage connoté est l'humus sur lequel se forme le langage conceptuel. Sans humus, pas de plante possible, mais l'humus n'est pas lui-même une plante. De même, sans langage connoté, pas de langage conceptuel et donc pas d'action possible ; mais le langage connoté ne peut pas nourrir directement l'action.

Certaines personnes, attachées à la fécondité du langage connoté et sensibles à la richesse du monde qu'il permet de représenter (l'allusion poétique comble les lacunes du langage comme la succession rapide des images au cinéma crée la sensation du continu) refusent la « sécheresse » du langage conceptuel ; ce faisant elles se mutilent du côté de l'action volontaire et se limitent à un rôle contemplatif. Certes ce rôle peut apporter des plaisirs esthétiques, mais non les plaisirs et leçons que procure l'action.

D'autres personnes, attachées à des finalités pratiques et éprises d'efficacité, refusent au contraire l'ambiguïté du langage connoté et ne veulent utiliser que le langage conceptuel. C'est souvent le cas des ingénieurs et des informaticiens. Ils en viennent à se couper des autres, auxquels ils parlent avec la même rigueur formelle que s'ils écrivaient un programme.

Ainsi les ingénieurs font des reproches à ceux qu'ils qualifient de « littéraires »

inefficient uses of computing machinery » (Knuth, [51] vol. 1, p. 170).

(philosophes, sociologues, historiens et autres « poètes ») et certains sociologues, philosophes etc. exècrent les ingénieurs, les « techniciens » dont ils dénoncent la « froideur inhumaine » et le « technicisme ».

Les critiques adressées à la technique paraissent cependant étranges si l'on convoque l'étymologie : *τέχνη* veut dire « savoir-faire ». Comment pourrait-on être « contre » le savoir-faire, le savoir pratique, l'efficacité ? Ce n'est pas le savoir-faire que visent en fait les adversaires de la technique, mais le langage conceptuel, la modélisation qui rend compte du monde de sorte que l'on puisse agir sur lui ; ils visent la déperdition symbolique, la perte des qualités allusives du langage dont on doit payer la modélisation ; ils visent aussi les attitudes « froides », « inhumaines » de ceux qui se vouent au langage conceptuel.

Ils voudraient que l'on pût être pratiquement efficace tout en conservant dans l'action la richesse des connotations, l'ambiguïté suggestive de la langue : mais cela, c'est impossible.

Les disputes entre « scientifiques » et « littéraires » trahissent une incompréhension envers la respiration de la pensée. Celle-ci a besoin tantôt d'élargir la sphère de ses représentations, et pour cela de laisser aller les associations d'idées qui forment son terreau ; tantôt de construire, sur la base ainsi élaborée, des concepts et structures hypothético-déductives : pour cela elle doit se fermer aux sirènes de l'allusion et éliminer les connotations. Ne vouloir admettre que l'une ou l'autre des deux phases de la démarche, c'est comme dire que dans la respiration seule l'inspiration serait légitime, l'expiration étant à proscrire (ou l'inverse). Celui qui applique une telle règle sera bientôt étouffé.

Le flux qui renouvelle et alimente notre pensée passe par le langage connoté, et le langage conceptuel permet de mettre en exploitation le stock des représentations ainsi accumulées. Il n'existe pas de stock sans flux qui l'alimente, et le flux se perd s'il n'alimente pas un stock.

1.4.3 « Objet »

Dans un « langage à objets » (on dit aussi « langage orienté objets », on appelle « objet » un petit programme qui contient :

- le nom propre (ou matricule, ou identifiant) qui désigne sans ambiguïté un « individu » du domaine étudié (un client, un produit, un établissement, une machine, une pièce détachée etc.) ;

- diverses variables observées sur cet individu et dont il a été jugé utile de noter la valeur en mémoire (par exemple date et lieu de naissance, adresse et numéro de téléphone d'une personne ; adresse, activité principale, taille d'un établissement etc.) : on appelle ces variables « attributs » ;

- diverses fonctions qui, appliquées aux « attributs », lancent des traitements produisant d'autres attributs ou encore des messages d'erreur ou d'anomalie (calculer l'âge d'une personne à partir de sa date de naissance et de la date du jour ; mettre à jour la valeur d'un attribut à partir d'une nouvelle saisie ; s'assurer que la saisie est réalisée dans un format conforme, que la donnée a une valeur acceptable etc.) ; on appelle ces fonctions « méthodes » et elles transcrivent des « règles de gestion ».

Ainsi l'objet (informatique) représente une entité du monde réel ; il garde trace de certains de ses attributs (mais non de tous, car toute entité du monde réel possède

une infinité d'attributs) ; il leur associe des traitements spécifiques.

Le plus souvent, l'objet est un cas particulier au sein d'une « classe » : l'objet qui représente *un* client est un cas particulier au sein de la classe « client ». Lorsque l'on définit une classe, on définit la liste des attributs et méthodes que l'on veut connaître sur chacun des individus de cette classe. Lorsque l'on indique les valeurs prises par l'identifiant et les attributs pour un individu quelconque, on dit que l'on « instancie » la classe dont l'objet particulier constitue une « instance ».

Ce jargon s'éclaire si l'on pense à ce qui se passe lorsque l'on fait une enquête statistique (Volle [85]). L'individu appartenant au champ de l'enquête, c'est l'entité qu'il s'agit de représenter. Le dessin du questionnaire, c'est la définition de la classe. Remplir le questionnaire, c'est l'« instancier » pour représenter un individu particulier. Les règles de codage et de vérification automatique utilisées lors de la saisie sont des « méthodes » au sens des langages à objets.

Mais « objet » est un faux ami : lorsque l'informaticien l'utilise pour désigner la représentation d'un existant, il s'écarte de l'usage courant comme de l'usage philosophique où « objet » désigne un *existant* repéré par la perception ou visé par l'intention d'un sujet. Cela conduit l'informaticien à prononcer une phrase qui fait se hérissier les cheveux du philosophe : « un objet, c'est une abstraction ». C'est presque une tautologie si l'on sait que le terme « objet », pour l'informaticien, désigne une représentation : toute représentation résulte d'une abstraction. C'est une absurdité si l'on donne au terme « objet » le sens qu'il a dans le langage courant comme en philosophie.

1.4.4 « Donnée » et « Information »

« Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem ». (Shannon, [75] p. 31.)

La quantité d'information qu'apporte un texte serait, selon la « théorie de l'information » de Shannon, d'autant plus grande que le texte est plus long et moins redondant. Une suite de lettres tirées au hasard ne comporte aucune redondance ; elle contiendrait donc, si l'on veut la reproduire exactement, plus d'information (au sens de Shannon) que n'importe quel texte de même longueur. « Wan trus be lifx », tapé au hasard sur mon clavier, contiendrait plus d'information que la phrase de même longueur « le pape est mort ».

Pour un lecteur il n'en est évidemment pas de même ; la seconde phrase a un sens alors que la première n'en a aucun. C'est que Shannon pensait non à l'être humain, mais à l'ordinateur et aux réseaux. Sa théorie est non pas une « théorie de l'information », mais une « théorie des données » ou mieux une « théorie des télécommunications » ; l'expression « théorie de l'information » suscite des contresens qui empêchent de distinguer deux concepts également utiles¹².

12. Pour une analyse de la théorie de Shannon, voir Escarpit [22]. L'article fondateur de Shannon n'a d'ailleurs pas pour titre « théorie de l'information », mais « théorie de la communication » (Shannon [74]).

* *

Ce qui nous est *donné*, c'est le monde de la nature et l'expérience que nous en faisons ; ce que l'on appelle « donnée » en informatique (comme en statistique), c'est le *résultat* d'une observation faite sur l'un des êtres qui existent dans ce monde, autrement dit la mesure d'une variable sur un individu.

Cette mesure n'est donc pas *donnée*, mais *obtenue* à l'issue d'un processus d'abstraction qui comporte plusieurs étapes : nous avons décidé (1) d'observer telle population parmi celles que comporte le monde de la nature, (2) d'observer telle sélection de variables sur les individus qui la composent, (3) de coder chaque variable de telle façon (unité de mesure, format de la donnée, nomenclature pour une variable qualitative), (4) d'identifier tel individu au sein de cette population, enfin (5) d'observer sur cet individu la valeur des variables sélectionnées.

Certaines personnes oublient cependant que les « données » résultent d'une observation sélective et les prennent pour la reproduction fidèle et complète de la réalité. D'autres estiment par contre que, du fait de cette sélection, les données seraient irrémédiablement « subjectives », et donc douteuses : pourtant les valeurs qu'elles prennent ne dépendent pas du bon vouloir de l'observateur... Il y a là un nœud de confusions à démêler.

* *

L'informatique enregistre, traite et stocke des données ; l'utilisateur les saisit ou les consulte et lance des traitements qui produisent d'autres données. Les flux qui en résultent dans les processeurs et les réseaux, les stocks qui s'accumulent dans les mémoires, les délais de mise à disposition, le synchronisme des répliquions, la concurrence entre des mises à jour simultanées, tout cela pose des problèmes de physique, de la « physique des données ». C'est le domaine propre de la technique informatique. Elle décide du dimensionnement des ressources (débit des réseaux, puissance des processeurs, taille et délai d'accès des mémoires) qu'elle organise selon l'ordre des performances et des coûts : les mémoires d'accès rapide, coûteuses, seront de faible volume et réservées aux utilisations urgentes, le gros des données est stocké sur des mémoires peu coûteuses à l'accès plus lent. Le système d'exploitation transfère automatiquement les données entre les divers types de mémoire. Le réseau est dimensionné par arbitrage entre le coût du haut débit et le besoin de transferts volumineux et rapides etc.

Les données sont organisées selon des architectures diverses : ici l'on a réalisé un système en mode objet, là une base de données relationnelles ; ici on a utilisé telle nomenclature ou tel typage, là une autre nomenclature, un autre typage. La communication entre les diverses parties du système d'information demande des transcodings et restructurations qui sont effectués par des *interfaces*. Savoir définir ces architectures, savoir interpréter les offres des fournisseurs de solutions, c'est là un métier de spécialiste.

Les données sont parfois fausses : des erreurs se produisent lors de la saisie ne serait-ce qu'en raison des fautes de frappe. C'est pourquoi les doubles saisies, les recopiations de données à la main, sont un des points de fragilité du système d'information. Par ailleurs lorsqu'une nomenclature change il est parfois nécessaire de procéder à la correction rétrospective des données, et le plus souvent cette correction

se paie par des inexactitudes.

* *

Prenons le mot « information » non au sens qu'il a dans la théorie de Shannon, ni à celui qu'il a dans le langage courant (les « informations de 20 heures »), mais au sens étymologique : une information, c'est quelque chose qui vous *in-forme*, qui modifie ou complète la *forme* intérieure de votre représentation du monde, qui vous *forme* vous-même¹³. L'information ainsi conçue a une *signification* : elle suscite une action de la part de celui qui la reçoit, ou du moins elle modifie (*trans-forme*) les conditions de son action future.

Pour pouvoir recevoir de l'information, il faut avoir été formé ; et c'est en recevant de l'information que l'on se forme. Certes il faut une amorce pour ce cycle, mais elle est enfouie dans les origines de la personne tout comme l'amorce du cycle de la poule et de l'œuf est enfouie dans les origines de la vie.

Une donnée ne peut donner naissance à une information que si elle est communiquée au destinataire dans des conditions telles qu'il puisse *l'interpréter*, la situer dans son propre monde intérieur et lui attribuer un sens. C'est là le but implicite de nos bases de données, de nos systèmes d'aide à la décision. Le statisticien connaît le travail qu'il faut fournir pour interpréter les données : les confronter avec d'autres, les organiser en série chronologique, évaluer des corrélations, revenir sur les définitions et conditions de l'observation etc. Il se publie beaucoup de tableaux de nombres mais peu de personnes disposent de l'arsenal nécessaire pour les interpréter. En fait, il faut le dire, personne ne les regarde sauf s'ils sont accompagnés de la synthèse en langage naturel qui permet de les faire parler¹⁴.

Mais les données ne sont pas utilisées principalement pour produire des statistiques : elles servent surtout à traiter des cas particuliers. Des décisions concernant chacun de nous sont prises à partir de dossiers où nous sommes représentés par quelques données plus ou moins bien choisies, plus ou moins exactes. L'agent qui contrôlera les traitements automatisés et traitera notre cas pourra-t-il transformer ces données en informations pour comprendre notre situation ? Ou bien se comportera-t-il en assistant de l'automate ?

* *

Le mot « comportement » nous fait passer de la physique des données à la *physique de l'information*. L'utilisateur d'une information *se comporte*. Il n'est pas une chose qui obéirait aux lois de la physique comme le fait le paquet d'octets qui transite par un réseau : il agira en fonction de ce qu'il a compris. La physique de l'information ressemble à celle de la circulation routière où les conducteurs, eux

13. « Informer » et « instruire » sont très proches : informer, c'est donner une forme ; instruire, c'est donner une structure.

14. On a pu définir ainsi la fonction de la synthèse statistique : « consentir une perte en information (au sens de Shannon) pour obtenir un gain en signification (information au sens étymologique) ». Le travail du statisticien est semblable à celui du typographe. La typographie fait perdre l'information que contiennent l'écriture manuscrite, les corrections etc. ; mais la mise en forme du texte imprimé facilite sa lecture et aide au dégagement du sens par le lecteur. De même la synthèse statistique attire l'attention sur les données significatives, ce qui invite à négliger celles qui ne le sont pas : cela prépare les voies de l'interprétation (Volle, [86] p. 51).

aussi, se « comportent » :

1) La route qui relie telle banlieue au centre ville est encombrée, on décide de l'élargir : elle sera tout aussi encombrée car comme elle est plus large davantage de personnes prendront leur voiture. De même, vous dimensionnez largement le réseau pour faciliter la tâche des utilisateurs, le nouveau réseau sera tout aussi encombré car de nouveaux usages s'y installeront.

2) Vous affichez sur un panneau de l'autoroute « bouchon à 6 km » ; certains conducteurs prendront la prochaine sortie et viendront encombrer le réseau des routes secondaires, d'autres resteront sur l'autoroute : ce choix est aléatoire. De même, vous avez installé dans votre entreprise un routeur de messages protégé par un pare-feu modeste : des pirates sauteront cette barrière pour utiliser votre routeur gratuitement. Vous augmentez la puissance du pare-feu : vos utilisateurs seront gênés etc.

La conception d'un système d'information doit anticiper le comportement des utilisateurs, tout comme ceux qui conçoivent un réseau routier anticipent le comportement des conducteurs. Le système d'information influence en retour le comportement de ses utilisateurs. Vous avez organisé de telle façon l'annuaire de l'organisation, découpé de telle sorte les zones géographiques, choisi telle nomenclature de produits : c'est ainsi que votre entreprise parlera, s'organisera, communiquera. Certaines décisions, prises à la va-vite par un groupe de travail, détermineront ainsi à long terme le cadre des représentations selon lesquelles l'entreprise définira ses priorités. La délimitation des populations décrites dans le système d'information, la gestion des identifiants, des nomenclatures, des classes d'objets, bref l'ensemble des opérations que l'on a coutume de nommer « administration des données » ou « gestion du référentiel », conditionnent la physique de l'information.

1.4.5 « Numérique »

Que d'émotions autour du mot « numérique » ! La « fracture numérique » mettrait en péril la cohésion sociale ; l'ordinateur constituerait un danger majeur pour les arts qu'il priverait de leur âme en les « numérisant ». Un frisson parcourt l'échine du « littéraire » confronté à la « froideur » des mathématiques et de l'abstraction.

Chacun est certes libre de ses goûts et dégoûts, mais on ne peut admettre que s'installe une erreur de jugement fondée sur un pur effet de vocabulaire. Nous devons donc développer ici l'analyse d'une erreur triviale : des esprits par ailleurs distingués la commettent. Comme ils sont éloquents et écoutés, il en résulte dans les idées un désordre dévastateur¹⁵.

La confusion est pire encore lorsque l'on utilise le terme anglais « digital ». « Digit » signifie « chiffre » en anglais mais « digital » désigne en français ce qui est relatif au doigt. L'expression « son digital » ahurit le badaud - c'est sans doute le but visé par des vendeurs péremptoires, fût-ce au prix d'une détérioration de la langue.

Il est vrai que dans l'ordinateur, au cœur du processeur qui effectue les opéra-

15. Virilio [84] annonce « la fin du langage ». Heureusement cette prophétie effrayante n'est étayée par aucun argument, pas plus que la dénonciation du « cybermonde » que Virilio fonde sur la ressemblance des mots « interactivité » et « radioactivité ». Certains voient là de la profondeur.

À propos de la « fracture numérique »

Cette expression, aussi disgracieuse que « son digital », désigne la différence sociale qui s'instaurerait entre ceux qui maîtrisent l'ordinateur et ceux qui, ne le maîtrisant pas, risqueraient l'exclusion.

Certains de ceux qui manifestent ainsi leur souci envers les exclus potentiels se flattent de n'avoir aucune pratique de l'ordinateur. Sans doute se considèrent-ils comme des privilégiés au grand cœur, préoccupés par l'inégalité qui les sépare de leurs inférieurs, inégalité qu'ils savourent tout en la déplorant. Comme il est délicieux de gagner à la fois sur le tableau social et sur le tableau moral ! Cela rappelle les « préceptes orgueilleusement humbles d'un snobisme évangélique » que la princesse de Parme inculque à sa fille dans *À la recherche du temps perdu*^a. Pourtant personne, quel que soit son niveau social, n'est embarrassé pour utiliser un distributeur automatique de billets ou un téléphone portable, outils « high tech » s'il en est. Les assistantes sont plus expertes que leur patron dans l'utilisation de l'ordinateur. Les personnes les plus calées en informatique, celles qui maîtrisent les langages de programmation et les architectures (savoir qui demande des années de formation), sont des cadres moyens à qui leur spécialité procure une légitimité qui leur est refusée par ailleurs. Les cadres supérieurs, sauf exception, ne feront pas l'effort de se qualifier en informatique tant que leur légitimité ne sera pas écornée par leur incompetence, et cette heure-là n'a pas encore sonné en France.

La vraie « fracture numérique » se trouve entre les dirigeants et l'ordinateur, et non tout près de la frontière de l'exclusion sociale. Bien sûr personne ne pourra utiliser un ordinateur sans un minimum d'explications et de pratique ; mais il ne faut pas prétendre que certains souffriraient, à cet égard, d'un handicap social : il est beaucoup plus difficile d'apprendre à parler en bon français que d'apprendre à utiliser l'ordinateur (l'apprentissage de la programmation, lui, est par contre aussi difficile que celui d'une langue naturelle).

^a « Rappelle-toi que si Dieu t'a fait naître sur les marches d'un trône, tu ne dois pas en profiter pour mépriser ceux à qui la divine Providence a voulu (qu'elle en soit louée !) que tu fusses supérieure par la naissance et par les richesses. (...) Sois secourable aux malheureux. Fournis à tous ceux que la bonté céleste t'a fait la grâce de placer au-dessous de toi ce que tu peux leur donner sans déchoir de ton rang, c'est-à-dire des secours en argent, même des soins d'infirmière, mais bien entendu jamais d'invitation à tes soirées, ce qui ne leur ferait aucun bien, mais, en diminuant ton prestige, ôterait de son efficacité à ton action bienfaisante. » (Marcel Proust, *À la recherche du temps perdu*, *Le côté de Guermantes* II, 1921 ; Robert Laffont, collection « Bouquins » 1987, vol. 2 p. 352).

tions, n'existent que deux niveaux d'une tension électrique qui servent à coder des 0 et des 1, des « bits ». L'information que l'utilisateur traite (texte, images, calculs, sons) est transcrite par une cascade de codages qui la traduisent ou l'interprètent pour parvenir au microcode, écrit en bits, que le processeur pourra exécuter.

Ce codage a-t-il une influence sur l'information ? Non, il ne fait que la transcrire. Si je tape une fable de La Fontaine sur mon clavier, le texte de la fable s'affichera sur l'écran avec toute sa puissance évocatrice ; le fait que les caractères soient codés en octets (huit bits) n'enlève rien à son contenu ; les conventions de traitement de texte que j'utilise pour le mettre en page, elles aussi transcrites en bits, ne font que faciliter sa lecture.

L'ordinateur, faisant ici fonction de machine de traitement de texte, est bien sûr incapable d'interpréter le texte mais il aide à sa présentation. On ne peut pas dire que la fable de La Fontaine soit « numérisée » : elle reste un texte poétique qui vise, par ses suggestions et sa musicalité, à émouvoir le lecteur et éveiller son intelligence.

Parodions, en le transposant, le raisonnement des critiques du « numérique »¹⁶ : « Comment voulez-vous que l'écriture puisse reproduire la richesse et les nuances du langage humain ? Comment pourrait-on décrire des couleurs quand on écrit en noir sur du papier blanc ? » etc. Cette transposition met à nu le procédé qu'utilisent des sophistes pour susciter doute et perplexité : on feint de croire, en confondant les diverses couches du processus, que la physique du support rétroagit sur le contenu du texte. Quand Marshall McLuhan dit « the medium is the message » (McLuhan, [59]), cela ne signifie pas « les ondes électromagnétiques constituent le message de la télévision », mais « les conditions économiques, sociales, de la production des programmes télévisuels ont sur leur contenu une influence qui peut être déterminante », ce qui est tout différent.

C'est en considérant les conditions pratiques, sociales, culturelles, économiques de la mise en œuvre de l'APU, de l'*utilisation* de l'ordinateur, que l'on peut raisonner sur ses apports, leurs limites et leurs dangers. Le mot « numérique » ne donne pas la clé de ce raisonnement : comme un épouvantail, il inhibe le discernement. Des personnes par ailleurs compétentes raisonnent mal quand elles parlent de la numérisation¹⁷ : elles ne voient pas que l'architecture en couches de l'ordinateur implique une différence de nature entre ce que fait l'utilisateur et ce que fait la machine. Que penseraient-elles si on disait de leurs écrits « ce ne sont que des signes noirs dessinés sur un papier blanc », ou de leurs paroles « ce ne sont que des ondes sonores que propagent des variations de la pression de l'air » ?

16. Leur voix, quand ils téléphonent, est soumise au codage MIC qui la transforme en un flux de 64 000 bits (zéro ou un) par seconde. Il en est de même, avec un débit plus élevé, pour la musique lorsqu'ils écoutent un enregistrement sur disque compact. Leur parole, la musique, sont-elles pour autant « numériques » ?

17. « L'ordinateur est la matérialisation de la logique mathématique : ils ont connu des développements historiques conjoints. Aux fondements de ces développements se trouve le principe d'identité. L'ordinateur calcule 0 et 1 mais ne sait faire que cela. Tout, en effet, est ramené à des 0 et des 1 afin que le courant électrique passe ou ne passe pas. L'ordinateur oblige à faire des modèles entièrement logiques. Il fonctionne comme un principe de réalité technico-logique, garant de la cohérence des modèles : un producteur de modèles hyperrationnels. » (Francis Pavé, « Transformation des représentations et résistance aux changements », conférence à l'école d'été 1998 de l'IUFM de Franche-Comté). Mais quel est le modèle « hyperrationnel » à l'œuvre quand on fait du traitement de texte ?

L'ordinateur est pour nos sociétés une innovation aussi importante que le furent, à d'autres époques, l'invention de l'écriture ou de l'imprimerie : il modifie les conditions de création, classement, recherche et traitement des données et documents. Il ne sera pas facile d'apprendre à s'en servir, à éviter ses effets pervers. Mais avons-nous vraiment maîtrisé l'écriture, l'impression ? Savons-nous vraiment lire et écrire (et compter) au sens non de la performance, mais de la vie intellectuelle et de la vie sociale ? faut-il donc que l'ordinateur nous inspire plus de craintes que la presse, les médias ou même la parole ?

1.5 Modèle en couches

Le « modèle en couches » a été conçu par des techniciens pour pouvoir penser un objet réel dont le fonctionnement articule plusieurs logiques. Ainsi dans un ordinateur se produisent des phénomènes électromagnétiques (des électrons se déplacent, des tensions se modifient) ; ces phénomènes sont traduits en 0 ou 1 ; les suites binaires représentent des instructions ou des données ; les règles qui gouvernent le codage des données et instructions sont fournies par les langages de programmation ; le langage le plus proche de la machine étant incommode pour l'être humain, des « langages de haut niveau » sont élaborés pour faciliter la programmation ; enfin, des « applications » sont programmées pour satisfaire les utilisateurs. Le mouvement des électrons ne peut pas être décrit selon la même grille conceptuelle que les applications, ni que le « compilateur » qui traduit un langage de haut niveau en instructions exécutables, mais tous ces phénomènes jouent conjointement (Tanenbaum [80]).

Le modèle OSI de l'ISO¹⁸, représente en sept couches le fonctionnement d'un réseau de télécommunications, de la couche physique (transport des bits) jusqu'à la couche « application » (Tanenbaum [81]).

Le modèle en couches ne s'applique pas qu'à la seule technique : des philosophes l'ont utilisé. Saussure ([18] p. 27) a décrit les couches psychologique, linguistique, neurologique, phonétique, sonore etc. par lesquelles passe la conversation entre deux personnes. On peut citer aussi Karl Popper et son « modèle des trois mondes » qui distingue et articule le « monde 1 » des objets et forces physiques, le « monde 2 » des états mentaux et le « monde 3 » des symboles, théories, problèmes, langages etc. (Popper [67] ; traduction : *La connaissance objective*, Flammarion 1998, p. 181). Maurice Blondel évoque un modèle en couches lorsqu'il dit, dans une phrase d'une remarquable densité, « entre (la science, la morale et la métaphysique), il n'y a point de contradiction, parce que là où l'on a vu des réalités incompatibles il n'y a (...) que des phénomènes hétérogènes et solidaires » (Blondel [7] p. XXII).

* *

Michel Foucault a décrit l'événement qui, à la fin du XVIII^e siècle, marqua le passage de la représentation classificatrice à une représentation *organique* de la nature : « Cuvier fera main basse sur les bocaux du Muséum, il les cassera et disséquera toute la grande conserve classique de la visibilité animale (Foucault [35] p. 150.) ».

18. « Open Systems Interconnection » publié de 1977 à 1986 par l'International Standard Organization.

La classification des êtres vivants est une opération ensembliste : l'évaluation des ressemblances et différences permet de définir une « distance » puis une relation d'équivalence entre espèces, les espèces « semblables » étant celles qui sont proches selon la distance choisie (Lecointre et Le Guyader [27]).

L'approche organique considère par contre non des collections d'êtres semblables, mais l'articulation des divers organes qui remplissent des fonctions complémentaires dans un même organisme. Si l'on peut considérer le corps d'un animal comme un ensemble de cellules, il sera souvent plus pertinent de le représenter comme un organisme : l'appartenance d'un bras ou d'un poumon au corps d'un être humain, l'appartenance de la direction commerciale ou de la DRH à une entreprise, ne sont pas de type ensembliste (ce ne sont pas des inclusions), mais de type organique (l'organe a une fonction spécifique, en relation avec les fonctions des autres organes). De même on peut certes voir dans un système d'information un ensemble de lignes de code source, mais il sera souvent plus pertinent de se le représenter comme un organisme.

Le modèle en couches, qui articule des logiques différentes, relève de l'approche organique. Bien qu'il soit d'origine technique il constitue une innovation philosophique. Il ne semble pas cependant que les philosophes aient vu dans ce modèle un outil au service du travail intellectuel, même si quelques-uns d'entre eux l'utilisent à l'occasion sans le savoir. Ce modèle est à l'œuvre, plus généralement, dans la pensée quotidienne. Ainsi dans un paysage comme *La montagne Sainte-Victoire vue de Bellevue*, de Cézanne, on voit une montagne (géologie) ; des arbres (botanique) ; un pont, des routes, des champs cultivés (action humaine) ; des nuages (météorologie). Ce paysage articule ainsi des êtres qui relèvent chacun d'une logique et d'un rythme spécifiques. L'unité de temps est pour la géologie le million d'années, pour la couverture végétale le millénaire ou le siècle, pour les constructions humaines le siècle ou la dizaine d'années, pour le déplacement des animaux ou des nuages l'heure ou la journée. Il serait impossible de rendre compte d'un paysage selon une seule logique, si ce n'est celle de la superposition de toutes les logiques qu'il fait jouer simultanément : le modèle en couches permet de penser l'articulation de diverses logiques et de leurs rythmes.

1.5.1 Couches de l'ordinateur

Le processeur d'un ordinateur fonctionne très vite mais ne peut reconnaître et exécuter qu'un nombre limité d'instructions élémentaires comme « copier dans ce registre l'enregistrement stocké dans la mémoire à cette adresse », « additionner le contenu de ces deux registres et noter le résultat dans ce troisième », « vérifier si le contenu de ce registre est égal ou non à zéro », « vérifier si le contenu de ce registre est supérieur à celui de cet autre registre », « stocker le contenu de ce registre dans un enregistrement ayant cette adresse » etc.

L'ensemble de ces instructions constitue le *langage machine* que nous noterons L_1 . L_1 est pauvre ; ses instructions s'écrivent sous forme de 0 et de 1 et il est très pénible de programmer dans ce langage.

On écrit donc en L_1 un ensemble d'instructions plus riche et plus proche du langage naturel, et on construit à partir de ces instructions un nouveau langage L_2

(« assembleur¹⁹ ») plus facile à utiliser.

Il existe deux façons d'exécuter un programme écrit en L_2 :

- un programme écrit en L_1 (nommé traducteur ou compilateur, ces deux termes sont synonymes) remplace chaque instruction du programme en L_2 par la suite équivalente des instructions en L_1 (traduction ou compilation). L'ordinateur exécute ensuite le programme en L_1 ainsi obtenu.

- un programme en L_1 (interpréteur), après avoir lu une instruction en L_2 , exécute immédiatement la séquence en L_1 équivalente (interprétation). L'interprétation évite de générer un programme en L_1 mais son exécution est plus lente que celle du programme compilé en L_1 .

Du point de vue de l'utilisateur, l'ordinateur doté d'un traducteur ou d'un interpréteur obéit aux instructions écrites en L_2 aussi docilement que si elles étaient écrites en L_1 : l'ensemble constitué de la machine physique M_1 , des langages L_1 et L_2 et de l'interpréteur (ou du traducteur) de L_2 et L_1 constitue la « machine virtuelle » M_2 . Pour celui qui écrit un programme en L_2 , M_2 est aussi « réelle » que M_1 (et son usage est plus commode).

Il existe cependant des limites à la complexité acceptable pour un traducteur ou un interpréteur. L_2 , bien que plus commode que L_1 , peut ne pas être encore le langage le plus convenable pour programmer. On écrit donc en L_2 un ensemble d'instructions et on construit un nouveau langage plus commode, L_3 , qui définit la nouvelle machine virtuelle M_3 . Langages et machines virtuelles s'empilent ainsi jusqu'à la couche n (figure 1.8).

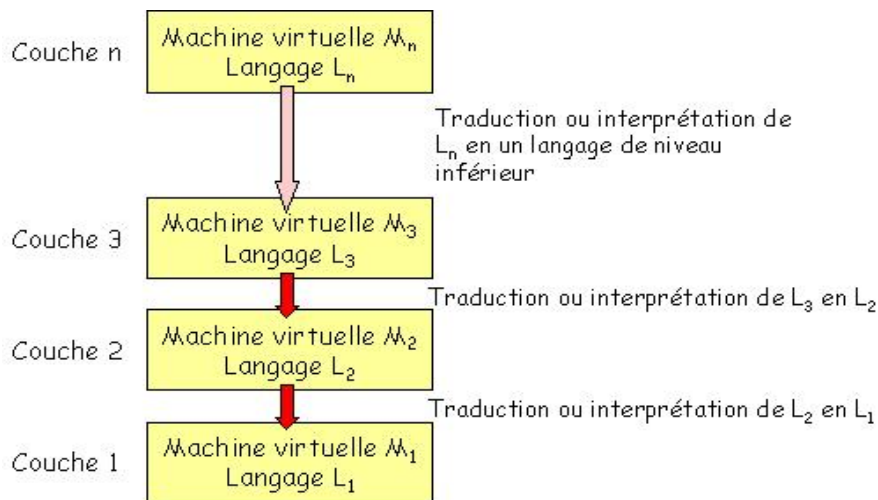


FIG. 1.8 – L'empilage des « machines virtuelles »

Dans les années 40, les ordinateurs n'avaient que deux couches (ou « niveaux ») :

19. Pour expliciter la façon dont la machine exécute les algorithmes et donner à ceux-ci la forme précise et complète que seule procure l'écriture d'un programme, Donald Knuth a défini l'ordinateur MIX (1009 en chiffres romains) et le langage assembleur MIXAL (Donald Knuth, [51] p. 124 et 144). Le lecteur peut ainsi, moyennant un effort d'attention, se représenter exactement la façon dont l'algorithme est exécuté et en déduire les limites de la précision du calcul, la durée d'exécution etc. Pour tenir compte de l'évolution des processeurs Knuth a aussi défini le MMIX (2009) qu'il a décrit sur son site www-cs-faculty.stanford.edu/~knuth/.

le niveau machine, dans lequel on programait, et le niveau physique qui exécutait les programmes. Les circuits de ce dernier niveau étaient complexes, difficiles à construire et peu fiables. Maurice Wilkes conçut en 1951 un ordinateur à trois couches afin de pouvoir simplifier le matériel. La machine disposait d'un interpréteur qui exécutait les programmes écrits dans le langage machine. Le matériel ne devait plus alors exécuter que des microprogrammes dont le répertoire d'instructions était limité, et non des programmes en langage machine.

Les assembleurs et compilateurs furent écrits dans les années 50. En 1954, le premier système d'exploitation (gestion automatique de la mémoire, gestion des répertoires de fichiers) fut conçu par Gene Amdahl pour l'IBM 704. La plupart des ordinateurs actuels possèdent six couches (figure 1.9).

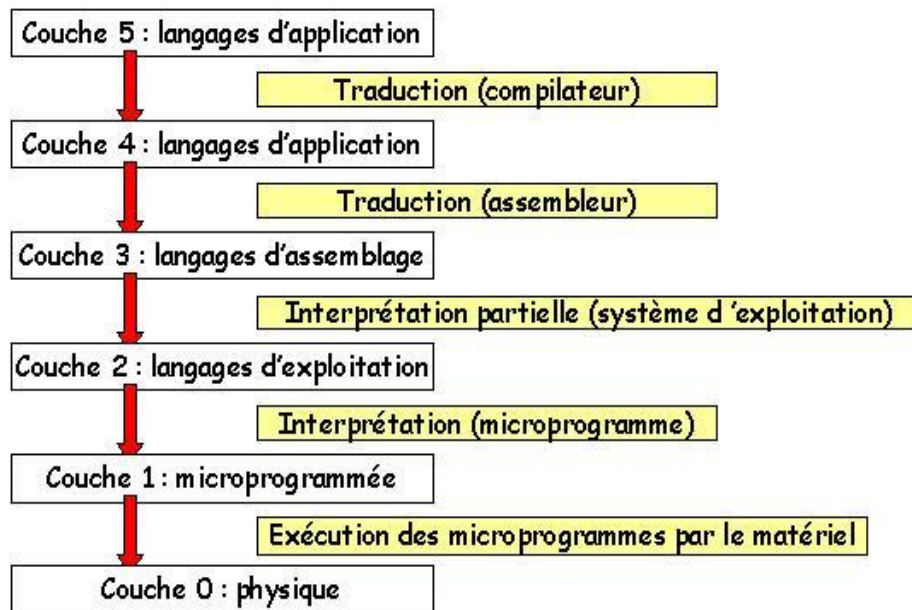


FIG. 1.9 – Couches de l'ordinateur

Quelqu'un qui utilise un ordinateur pour faire du traitement de texte met en œuvre un programme dont les instructions parviennent à la couche physique (processeur et mémoires) après une cascade de traductions et d'interprétations. L'utilisateur peut ignorer ces opérations : pour lui, et dans le cadre de cette application, l'ordinateur fonctionne comme une machine de traitement de texte et il n'a à connaître que cette seule machine (c'est pourquoi il est inexact de dire que son activité est « numérisée » même si l'ordinateur, pour exécuter ses ordres, les transcrit sous la forme de suites de 0 et 1).

Une instruction du niveau applicatif, traduite ou interprétée en cascade, engendre dans la couche microprogrammée de nombreuses instructions. Le temps d'exécution d'un programme peut s'évaluer en additionnant les temps unitaires : ce calcul est une étape importante de l'évaluation d'un algorithme (Knuth [51] vol. 1 p. 170).

Si l'on veut optimiser le délai de traitement il faut maîtriser la succession des instructions et pour cela programmer dans les « couches basses ». Cependant l'accroissement des performances des processeurs et de la taille des mémoires limite l'utilité de cette optimisation, sauf pour certaines applications où la rapidité est cruciale. Les programmes sont presque tous écrits dans des langages de niveau élevé ; l'art

du programmeur s'appuie alors sur une connaissance des propriétés du langage de programmation qui doit être suffisante pour anticiper les conséquences physiques des choix faits dans les couches hautes.

L'utilisateur d'un ordinateur n'a à se soucier que du service rendu par la couche dont il se sert et qu'il considère comme une ressource physique (« matériel et logiciel sont équivalents »). Celui qui conçoit un ordinateur optimise l'utilisation des ressources couche par couche, et non en considérant le processus d'ensemble (Kidder [50]) : la simplicité que l'on gagne ainsi fait plus que compenser une éventuelle perte en efficacité physique.

1.5.2 Portée du modèle en couches

Pour montrer à quel point le modèle en couches est présent (le plus souvent de façon implicite) dans notre vie quotidienne, considérons la conversation entre deux personnes (Saussure [18] p. 27). Elle emprunte les couches suivantes : conception et compréhension des idées à communiquer (logique) ; expression et décodage de ces idées dans un langage (sémantique) ; expression et décodage de ce langage dans des phonèmes (phonétique) ; mécanismes de l'articulation et de l'audition (physiologie) ; émission et réception d'ondes sonores (physique) ; émission et réception d'ondes sonores (physique).

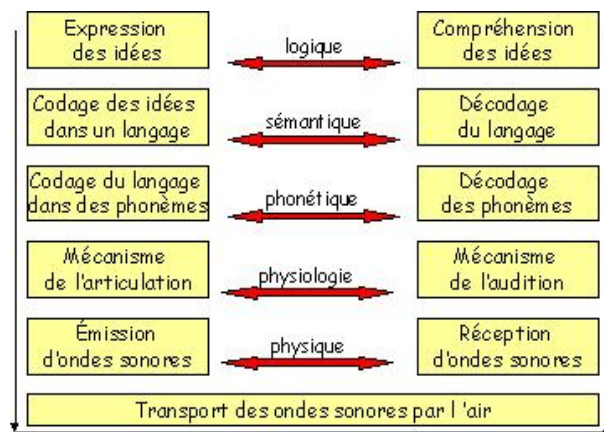


FIG. 1.10 – Couches de la conversation

Deux personnes qui conversent font fonctionner ces diverses couches alternativement dans les deux sens mais elles ne s'intéressent qu'aux idées qu'elles émettent ou reçoivent. Les couches inférieures n'attirent l'attention que si elles fonctionnent mal : si l'un des locuteurs parle une langue ignorée de l'autre, ou encore s'il est sourd ou aphasique ; les ondes sonores ne sont utilisables que si les deux interlocuteurs sont assez proches l'un de l'autre et si le milieu sonore ambiant n'est pas trop perturbé par le bruit.

Il suffit d'ajouter quelques couches en bas du modèle pour représenter une conversation téléphonique : l'onde sonore est codée sous la forme d'une onde électromagnétique par le poste téléphonique, numérisée dans le commutateur de départ, transportée par le réseau, décodée à l'arrivée.

Le modèle en couches décrit un empilage de conditions *toutes également nécessaires*. Or si, dans un produit qui comprend des 0 et des 1, tous les termes sont

égaux à 1 sauf un seul qui est égal à 0, le produit est nul. De même, lorsqu'un phénomène obéit à plusieurs conditions simultanément nécessaires elles doivent toutes être respectées pour que le phénomène puisse se produire. Dès lors il serait vain de se demander laquelle de ces couches est « la plus importante », de tenter d'établir entre elles une hiérarchie.

Le modèle en couches permet donc de faire l'économie de certaines réflexions stériles. On tente parfois de déterminer l'origine et le responsable de la réussite d'un processus : cet effort est vain si le processus a traversé plusieurs couches toutes nécessaires. Si le stratège doit concentrer son attention sur la question la plus importante du moment, il n'en doit pas moins rester attentif à la coopération de diverses couches indispensables au processus ; son aptitude à coordonner des activités diverses se fonde sur la connaissance de leur articulation. Cette coordination n'est pas mélodique (une seule ligne dont il faut suivre le déroulement dans le temps) mais polyphonique (plusieurs lignes qu'il faut suivre ensemble et qui s'articulent entre elles).

1.6 Vers l'ordinateur individuel

Le silicium (Si) est, après l'oxygène, le constituant le plus massif et donc le plus répandu de la Terre. Le silex, constitué de silice pure (dioxyde de silicium SiO_2), a servi il y a 2,5 millions d'années de matière première aux premiers outils, armes et parures fabriqués par *Homo habilis* en Afrique de l'Est (Gille [39]). Par la suite l'être humain a abondamment utilisé le silicium sous forme de céramique, de brique, puis de verre et de porcelaine.

Le silicium pur cristallisé, produit par l'industrie chimique à partir de verres de silice (quartz) sert aujourd'hui de matière première aux « puces » de la microélectronique.

Le silicium est ainsi intimement lié à l'histoire de notre espèce. Il est peu coûteux en raison de son abondance, et ses propriétés électroniques expliquent à la fois le tranchant des arêtes du silex et les performances des circuits intégrés.

L'exploitation des potentialités du silicium requiert la maîtrise de plusieurs techniques délicates. Elle a donc été progressive. Cette progressivité explique la « loi de Moore » (doublement des performances tous les 18 mois) qui elle-même explique pour partie la montée des performances et la baisse du prix des ordinateurs, l'émergence de l'ordinateur individuel, puis la pénétration et la généralisation de l'ordinateur dans les entreprises et la société.

* *

En 1947, trois chercheurs des Bell Labs d'AT&T (John Bardeen, Walter Brattain et Robert Shockley) découvrent le « transistor » (« transfer resistor »). Leur invention s'appuie sur le fait qu'il est possible de contrôler le flux d'électricité dans le germanium (qui sera remplacé ensuite par le silicium) en faisant en sorte que certaines zones soient conductrices et d'autres isolantes, d'où le terme « semi-conducteur ». Comparé au tube à vide, technique jusqu'alors dominante, le transistor se révélera plus fiable, moins consommateur d'énergie et susceptible de miniaturisation.

Les premiers transistors étaient faits à la main dans des conditions rustiques si

La conception d'un ordinateur

Tracy Kidder ([50]) décrit les étapes de la conception du mini-ordinateur Eclipse MV/8000 lancé par Data General en avril 1980.

Projet, réalisation, intégration, déboguage, sont autant d'aventures qui s'articulent lors de la conception sous une contrainte de délai toujours présente, toujours violée, toujours renouvelée. La mise au point des langages et celle du matériel se renvoient la balle ; des spécialités diverses, aux psychologies contrastées, s'entrecroisent. Assurer la compatibilité du produit avec la gamme de l'entreprise, le respect des interfaces, suppose des compromis fondés sur le sens du marketing et du possible technique.

Les hommes qui font ce travail sont des ingénieurs au sens complet du terme : ils réalisent de façon ingénieuse un travail d'ingénierie. Ils se trouvent au front de taille de la recherche pratique, au point où elle pose des problèmes intellectuels d'une extrême subtilité, où elle rencontre la limite de notre connaissance de la nature. Ce sont des rêveurs pratiques, des passionnés. L'argent n'est pas leur moteur, ils le disent avec insistance et s'en étonnent eux-mêmes : ils ressemblent plutôt à ces alpinistes qui se lancent des défis.

La limite qu'ils cherchent, ce n'est pas tant une limite individuelle qu'une limite humaine : ils veulent voir jusqu'où un être humain peut aller dans la clarté de la conception, la maîtrise de la complexité, la concentration de la pensée, la rapidité d'exécution, l'élégance de la démarche.

L'animateur de l'équipe, Tom West, manifeste son respect pour ses collaborateurs en étant froid et distant. « Pas de tapes dans le dos », dit-il : mais il leur délègue des responsabilités qu'ils n'auraient jamais pu avoir ailleurs et conduit en stratège l'affaire vers le succès en assumant le poids des reproches et des frustrations de l'équipe.

On trouve dans ce livre une excellente description intime de l'ordinateur, de la façon dont il est perçu, pensé, voulu par ceux qui le conçoivent. Les pages consacrées au microcode, à l'assembleur, à la conception des cartes, sont très pédagogiques. L'articulation des couches de l'ordinateur devient ainsi intuitive, palpable.

Les concepteurs de l'ordinateur en restent cependant à « la machine ». Ils ne s'intéressent pas à la diversité de ses utilisations potentielles (c'est une autre question, qui devra être traitée par d'autres personnes à un autre moment que celui de la conception), mais ils veillent à la doter des performances qui lui permettront d'être efficace dans la plus grande diversité d'utilisations possible.

Ils créent et balisent ainsi l'espace où devra se conduire une recherche postérieure à la leur : la recherche sur les usages, sur l'assistance qu'apporte « la machine » au travail humain. Le livre, qui s'arrête au moment où « la machine » quitte les mains des ingénieurs pour passer dans celles des vendeurs (la rencontre entre ces deux populations est décrite avec humour), ne considère pas cette deuxième phase de la recherche : elle se déroulera chez les utilisateurs, les clients, les intégrateurs.

Du processeur au microprocesseur

Le processeur est la partie centrale et aussi la partie la plus complexe d'un ordinateur. Il comprend les registres et la machine qui exécute les opérations commandées par le programme^a. Les ingénieurs qui conçoivent un processeur doivent faire des choix en tenant compte d'un grand nombre de facteurs économiques et techniques : il en est résulté une diversité de solutions qui a rendu les premiers ordinateurs mutuellement incompatibles. Pour un utilisateur, il était coûteux de changer de fournisseur car il aurait fallu réécrire les programmes afin de les adapter à une autre machine.

Le microprocesseur, processeur intégré sur une seule puce, a introduit un standard de fait : Intel vend à qui veut les acheter ses microprocesseurs qui deviennent alors de simples composants et dispensent le client des études les plus complexes et les plus coûteuses (Brulé [12] p. 282).

^a Chargement dans un registre d'une valeur contenue dans une mémoire, stockage en mémoire de la valeur contenue dans un registre, opérations arithmétiques sur les registres, calculs sur les adresses, comparaison entre une valeur en mémoire et une valeur dans un registre, sauts d'instructions etc.

on les compare aux « salles blanches » actuelles. Les rendements étaient bas (de 20 à 30 %) et les performances très variables. Les progrès techniques ont donc d'abord concerné la maîtrise du processus de production.

Pendant les années 50 une nouvelle industrie se crée. La fabrication des semi-conducteurs fait d'importants progrès : la *diffusion* consiste à diffuser des impuretés (« dopants ») sur leur surface, ce qui supprime le processus fastidieux d'ajout de diverses couches de matériaux isolants et conducteurs sur le substrat. Des *techniques photographiques* permettent de projeter sur le semi-conducteur le dessin de masques compliqués de telle sorte que la diffusion ne se produise que sur les surfaces souhaitées.

Le circuit intégré (circuit électronique comportant plusieurs transistors sur une même pièce de matériau semi-conducteur) est inventé en 1958 par Jack Kilby²⁰. Jean Hoerni, de Fairchild Semiconductor, introduisit en 1959 une autre innovation essentielle en observant que les techniques de diffusion et de photographie permettaient de se libérer des complications du transistor à trois dimensions et de dessiner des *transistors plans* (« planars »). Il devenait alors possible de faire les connexions électriques non plus à la main, mais en déposant par condensation de vapeur métallique un film sur les parties appropriées du semi-conducteur. Moore a fait de l'invention du planar le point de départ de la « loi de Moore ». Fairchild Semiconductor produisit le premier circuit intégré utilisant cette technique en 1961.

Ces techniques permettront de passer de la production manuelle à la production industrielle de série avec une meilleure qualité. Les progrès des processus de production et des techniques se sont ensuite poursuivis. Les méthodes photographiques

20. Le premier circuit intégré, présenté le 12 septembre 1958, comportait deux transistors montés sur une barre de germanium.

sont devenues de plus en plus précises grâce notamment à l'emprunt de techniques de photolithographie conçues pour l'imprimerie.

Les premiers circuits intégrés étaient des mémoires. Le microprocesseur, qui est avec l'ampoule électrique et le téléphone considéré comme l'un des plus grands apports de la technique américaine, sera inventé en 1971 par Intel.

1.6.1 Origines et évolution du microprocesseur

Intel a été créée en 1968 à Santa Clara, dans la Silicon Valley, par Robert Noyce, Gordon Moore et Andy Grove. Ces trois ingénieurs venaient de quitter Fairchild Semiconductor à la suite d'un désaccord sur sa stratégie.

Sur la base d'un business plan d'une page tapé à la machine par Robert Noyce (voir page 35), Art Rock, « venture capitalist » de San Francisco qui avait financé le démarrage de Fairchild et connaissait les qualités professionnelles de ces trois ingénieurs, avança les 2,5 millions de dollars nécessaires²¹.

Intel a d'abord produit des mémoires. Busicom, fabricant japonais de machines à calculer, lui demande en 1969 de mettre au point douze circuits intégrés pour assurer les fonctions de ses machines. Un ingénieur d'Intel, Ted Hoff, proposa de concevoir un circuit intégré programmable unique pour réaliser l'ensemble de ces fonctions. Noyce et Grove approuvèrent cette solution, ainsi que Busicom. Le travail fut réalisé en neuf mois sous la direction de Federico Faggin.

Le 4004 est annoncé en novembre 1971. *C'est le tout premier microprocesseur*. Il comprend 2300 transistors, tourne à 108 kHz et a autant de puissance de calcul que l'ENIAC. Il utilise un chemin de données à 4 bits (4 chiffres binaires sont traités en parallèle), ce qui le rend utilisable pour des automatismes ou des calculettes mais non pour construire un ordinateur.

Intel rachète pour 60 000 \$ les droits de Busicom sur le 4004 (peu après, Busicom fera faillite). Il faut trouver d'autres clients. Intel se lance dans une communication active pour convaincre la profession d'écrire des logiciels pour le 4004 et de l'utiliser pour des automatismes comme ceux qui assurent le contrôle des feux de circulation.

En avril 1972 Intel produit le 8008. C'est un processeur à 8 bits, ce qui permet à quelques pionniers de construire un ordinateur autour du microprocesseur. En 1973 sort le Micral de R2E ; la revue *Byte* le baptise en juin 1973 du terme « microcomputer ». Le micro-ordinateur est né, mais il ne rencontrera le succès commercial qu'avec l'Altair (1974) qui utilisera le 8080 d'Intel sorti en 1974 et surtout avec l'Apple II (1977) qui utilisera le 6502 de MOS Technologies, sorti en 1976.

En 1974 Motorola, entreprise d'électronique créée en 1928 à Chicago et qui s'est spécialisée dans les circuits intégrés, produit son premier microprocesseur, le MC6800 à 8 bits. Il comporte 4 000 transistors. Apple adoptera le processeur 16 bit 68000 de Motorola, sorti en 1979, pour le Lisa (1983) puis pour le Macintosh (1984).

Intel et Motorola produisent des microprocesseurs, mais ils ne sont pas les seuls :

21. Cet épisode est caractéristique des relations entre capital risque et ingénieurs dans la Silicon Valley. L'investisseur n'est pas seulement un financier : c'est un *entrepreneur*, un expert qui connaît les personnes et les techniques, raisonne sur le marché des produits et anticipe leur évolution (Aoki [2]). Pour un contraste avec le cas de la France, on peut lire les aventures de Guillaume Portes (un « Bill Gates français ») imaginées par Jean-Pierre Brulé ([12] p. 307).

on peut citer AMD (Advanced Micro Devices, Sunnyvale, Californie) qui produit depuis 1991 des clones de microprocesseurs Intel, ainsi qu'IBM qui produit des PowerPC. Certaines entreprises dites « fabless » (sans fabrication) conçoivent des microprocesseurs qu'elles font fabriquer sur plan par un « fondeur » : c'est le cas de Sun qui fait fabriquer le SPARC par Texas Instruments, de Silicon Graphics avec le MIPS etc.

Le Pentium 4 de 2000 possède 42 millions de transistors et tourne à 1,5 GHz. Il permet à l'utilisateur de produire des films, diffuser de la télévision sur l'Internet, communiquer par la voix et l'image, afficher des graphiques 3D en temps réel, coder de la musique au format MP3, et faire tourner simultanément plusieurs applications multimédia tout en restant connecté à l'Internet ²²

1.6.2 La loi de Moore

En 1965 Gordon E. Moore était directeur de la recherche et du développement à Fairchild Semiconductor. Préparant un exposé sur l'évolution des performances des mémoires, il constata que la capacité des « puces » avait à peu près doublé chaque année de 1959 à 1965. Il fit l'hypothèse que cette tendance allait se poursuivre. C'est cette hypothèse que l'on appelle « loi de Moore » ; elle était hardie, puisque Moore ne disposait que de cinq observations pour l'étayer.

Moore publia sa découverte dans « Cramming more components into integrated circuits », *Electronics*, 19 avril 1965 [60]. Cet article devenu célèbre a introduit dans la conception des circuits intégrés une « anticipation autoréalisatrice », comme disent les économistes : ceux qui conçoivent un nouveau circuit intégré se font un devoir de la respecter (Colwell [17]). Par ailleurs elle a encouragé les chercheurs à anticiper sur la croissance des performances et à concevoir des systèmes utilisant une puissance très supérieure à celle disponible lors de leurs recherches (Kay [49] p. 515), ce qui a suscité une accélération de l'innovation.

Voici la phrase clé de cet article : « The complexity for minimum components costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the long term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimal cost will be 65,000. »

Le raisonnement de Moore comporte deux étapes. Il examine d'abord la relation entre le coût moyen de production par composant et la complexité du circuit. Cette fonction est d'abord décroissante, puis croissante : il existe donc un niveau de complexité pour lequel le coût moyen par composant intégré sur le circuit est minimal. C'est ce niveau que des producteurs rationnels choisiront car il procure le meilleur rapport efficacité/coût. Ensuite, Moore constate que ce niveau optimal de complexité est multiplié chaque année par deux.

Moore ne dit pas que le coût de production des circuits intégrés restera stable dans le temps malgré l'augmentation de leur complexité. Rien, dans la loi de Moore, ne le garantit.

22. Sur les origines du microprocesseur, voir www.intel.com/intel/intelis/museum/exhibit/hist_micro/hof/hof_main.htm et www.motorola.com/content/0,1037,115-110,00.html

Business Plan d'Intel, 1968

The company will engage in research, development, and manufacture and sales of integrated electronic structures in order to fulfill the needs of electronic systems manufacturers. This will include thin films, thick films, semiconductor devices, and other solid states components used in hybrid and monolithic integrated structures.

A variety of processes will be established, both at a laboratory and production level. These include crystal growth, slicing, lapping, polishing, solid state diffusion, photolithographic masking and etching, vacuum evaporation, film deposition, assembly, packaging, and testing, as well as the development and manufacture of special processing and testing equipment required to carry out these processes.

Products may include diodes, transistors, field effect devices, photo sensitive devices, photo emitting devices, integrated circuits, and subsystems commonly referred to by the phrase "large scale integration". Principle customers for these products are expected to be the manufacturers of advanced electronic systems for communications, radar, control and data processing. It is anticipated that many of these customers will be located outside California^a

^aSource : www.intel.com ; les fautes d'orthographe figurent dans l'original.

Le coût de production des circuits intégrés dépend peu de la quantité produite : c'est une *production à coût fixe*. Le coût de mise en production croît avec le niveau de complexité (la conception de l'Itanium a coûté à Intel 2 milliards de dollars²³, et pour le mettre en production il faut encore construire une usine coûteuse). Le coût moyen de production résulte de la division de ce coût fixe par le nombre d'unités vendues ; le prix est égal au coût moyen augmenté de la marge que le fournisseur peut s'attribuer puisqu'il est en situation de monopole (ou d'oligopole si l'on suppose que le consommateur peut choisir entre Intel, Motorola, AMD etc.)

La stratégie des fabricants de microprocesseurs est donc subtile : ils doivent dégager assez de profit pour financer la recherche et la croissance, mais aussi pratiquer des prix assez bas pour que le débouché de leurs produits puisse croître. Or le marché des ordinateurs est capricieux : formé de l'addition d'un marché de premier équipement et d'un marché de renouvellement, il est sensible à la mode dans sa première composante et à la conjoncture dans la deuxième²⁴. Il faut cadencer l'innovation pour le relancer périodiquement. On peut vendre cher les premiers microprocesseurs d'une nouvelle série, puisqu'ils procurent un gain de compétitivité (physique ou médiatique, peu importe ici) aux ordinateurs qui les utilisent ; puis il faut baisser leur prix pour élargir leur pénétration. L'évolution du prix des microprocesseurs résulte de l'ensemble de ces phénomènes ; si sa modélisation comporte la loi de Moore, elle ne s'y résume pas.

En 1975, Moore réévalua le rythme de croissance : désormais elle procédait par doublement *tous les 18 mois* et non tous les ans (figure 1.11). Néanmoins elle restait exponentielle. Elle s'écrit donc, en notant n_t le nombre de composants sur une puce l'année t :

$$n_t = n_{1975} 2^{(t-1975)/1,5}$$

L'accroissement de la densité des composants permet d'augmenter les performances : lorsque la distance entre transistors se réduit la vitesse de traitement s'accroît et on peut introduire sur la puce des fonctions auparavant remplies par d'autres équipements comme la carte graphique, le modem ou le contrôle de la mémoire.

En 1995, Moore vérifia que la progression prévue avait bien été respectée. Cependant lors du forum des développeurs d'Intel de septembre 1997 il a déclaré que l'accroissement de la densité des microprocesseurs pourrait atteindre vers 2017 une limite physique, celle de la taille des atomes.

* *

L'évolution des performances que décrit la loi de Moore résulte de la mise en exploitation progressive d'un phénomène naturel. Lorsque la croissance s'arrêtera vers 2017 nous devrions, si la prolongation de la tendance se confirme, disposer de processeurs et de mémoires $2^{12/1,5} = 256$ fois plus puissants qu'en 2005. L'informatique sera donc qualitativement différente de celle que nous connaissons aujourd'hui.

Aurons-nous répondu à toutes les questions que pose l'utilisation de cette ressource ? C'est peu probable. Ces questions relevant de l'organisation des entreprises

23. Source : www.forbes.com/2001/05/29/0529intel_print.html

24. Tout marché de renouvellement (automobiles, équipement ménagers etc.) est le théâtre d'une amplification du mouvement conjoncturel, le consommateur retardant l'achat de quelques mois ou années en basse conjoncture, et achetant en masse en haute conjoncture (voir page 40).

Étapes de la production d'un microprocesseur

(Source : Malone [56])

La production des cristaux de silicium, matière première de base, suit un processus chimique dont le coût de revient est minime : à partir d'un germe de cristal, la silice hautement purifiée croît en un long cylindre à l'intérieur d'un creuset à haute température. Puis ce cylindre est découpé en tranches fines par une scie au diamant.

Les tranches de silicium sont ensuite soumises à des manipulations de type photographique qui gravent les couches incorporant la logique et la physique du circuit intégré. Le coût de production des circuits intégrés est donc essentiellement composé (a) du coût de conception des « masques » qui permettent le dessin des circuits, (b) du coût des équipements automatiques et hautement protégés (notamment contre les poussières) qui assurent la production. Il ne dépend pratiquement pas du nombre de circuits intégrés produits (la production est « à coût fixe »).

- 1) Chaque tranche est nettoyée à l'acide, puis placée dans un four à haute température où sa surface est oxydée.
- 2) La tranche est recouverte d'un produit photosensible et le dessin du premier niveau du circuit est projeté à travers un masque par un faisceau de rayons ultraviolets. Puis les surfaces impressionnées par ce faisceau sont enlevées par de l'acide. Le processus est répété pour chaque niveau du circuit. La tranche est traitée avec des impuretés chimiques positives ou négatives qui créent les zones conductrices. Finalement, elle est revêtue d'un enduit qui protège sa surface et empêche les fuites de charges électriques.
- 3) Les puces comportant des fautes sont repérées sur la tranche par inspection visuelle et test informatique. Puis les puces sont découpées et collées sur un support. De minuscules fils sont soudés pour connecter le support aux points de contact de la puce, un couvercle est placé sur la puce et scellé pour la protéger.
- 4) Le microprocesseur est mis pendant plusieurs jours dans un four à basse température pour simuler son utilisation à long terme. Il est ensuite testé et les microprocesseurs à basse performance sont éliminés.

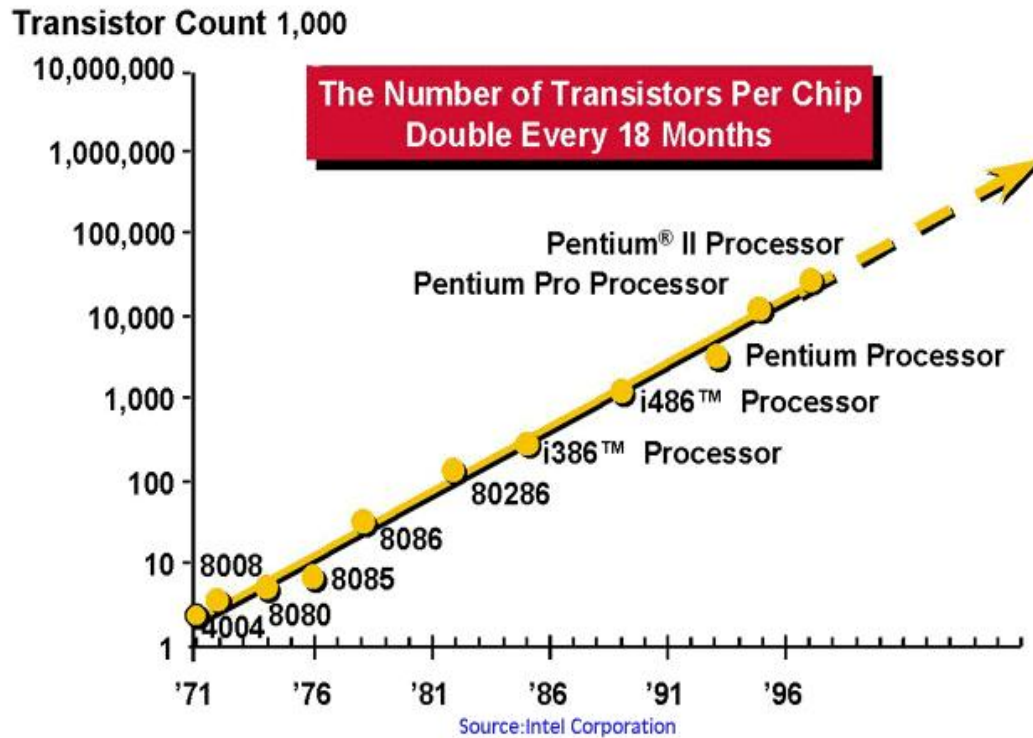


FIG. 1.11 – Illustration de la loi de Moore

et de la vie en société, elles concernent tout le monde et ne pourront donc pas se régler aussi vite que ne le font les questions techniques qui, elles, sont traitées par la sous-population restreinte des ingénieurs « pointus ».

L'utilisation de la ressource naturelle que constitue le silicium, matière peu coûteuse mais riche en potentialités, nous occupera pendant le XXI^e siècle et sans doute encore par la suite.

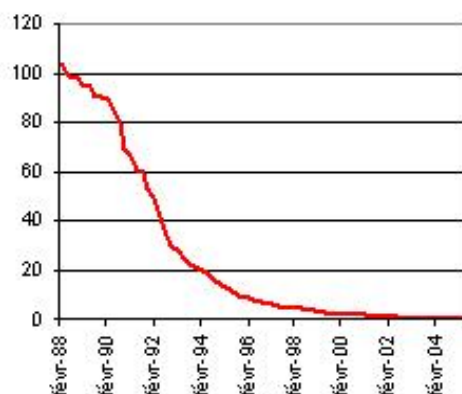
1.7 Évolution du prix des micro-ordinateurs

L'indice du prix des micro-ordinateurs est l'un des indicateurs les plus importants pour la compréhension de l'économie des TIC. L'INSEE le publie sur www.insee.fr, série PVIC300201. Il est disponible selon la périodicité trimestrielle depuis le premier trimestre 1988, selon la périodicité mensuelle depuis janvier 2003. Nous présentons ici une série longue composée en enchaînant les bases successives.

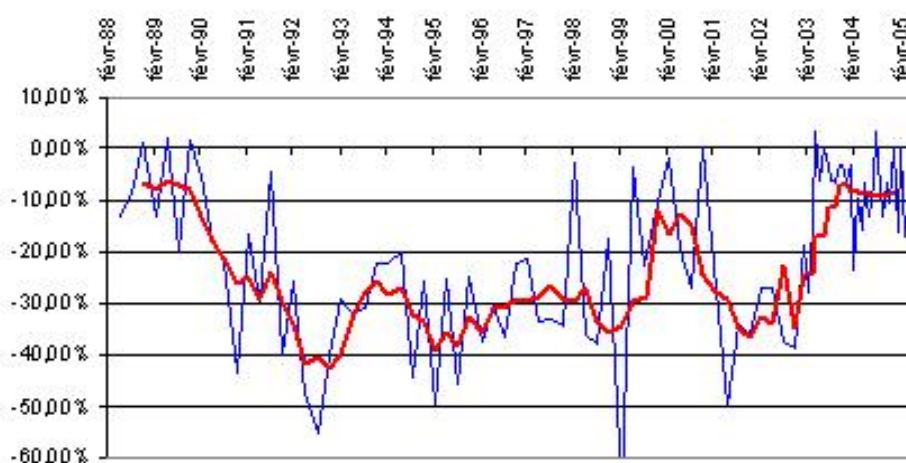
Il s'agit d'un indice hédonique, c'est-à-dire à qualité constante. La moitié environ de la baisse de l'indice s'explique par la baisse du prix moyen de vente des micro-ordinateurs, l'autre moitié s'explique par la hausse de la qualité.

La courbe de la figure 1.12 prend à partir de 1990 l'allure d'une exponentielle décroissante ; la valeur de l'indice est en octobre 2005 égale à 0,6 % de sa valeur au premier trimestre 1988.

Pour pouvoir l'interpréter, observons le taux de variation en équivalent annuel (le niveau « - 20 % » sur la figure 1.13 signifie que l'indice a évolué ce trimestre-là à un taux qui, sur un an, correspondrait à une baisse de 20 %). Dans le graphique,

FIG. 1.12 – *Indice du prix de vente industriel des micro-ordinateurs, INSEE*

le taux de variation trimestriel est en bleu et la moyenne mobile sur cinq trimestres est indiquée en rouge.

FIG. 1.13 – *Taux de variation annuel du prix des micro-ordinateurs*

On voit que la baisse annuelle a été durant les années 90 de l'ordre de 30 %. La rapidité de cette baisse explique la pénétration rapide des micro-ordinateurs dans les entreprises et dans le grand public. La baisse s'est ralentie dans les années 2000 : elle se poursuit depuis la mi-2003 au rythme annuel de 8,2 %.

* *

Le *Department of Labor* des États-Unis publie depuis décembre 1992 un indice mensuel du prix des ordinateurs de bureau et un autre pour les ordinateurs portables²⁵. La comparaison des taux de variation (en moyenne mobile sur cinq

25. (Source: Department of Labor « Producer Price Index », stats.bls.gov/ , Séries

trimestres) montre des évolutions parallèles, la baisse du prix des portables étant toutefois un peu plus rapide que celle des ordinateurs de bureau (figure 1.14).

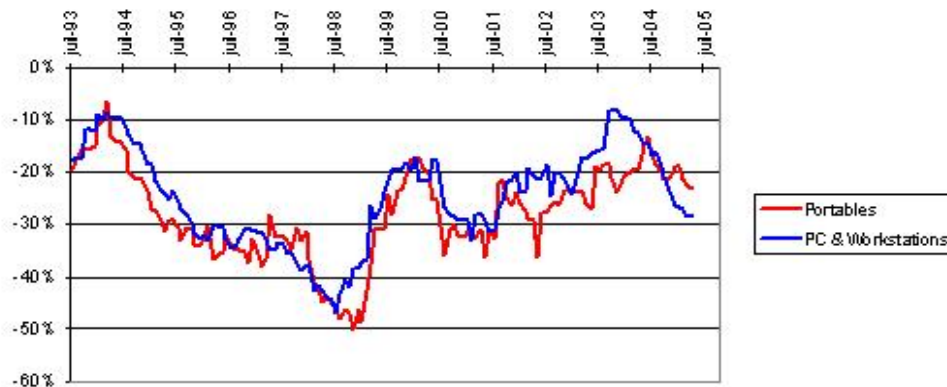


FIG. 1.14 – *Taux de variation des prix des ordinateurs de bureau et des ordinateurs portables aux États-Unis*

Si l'on compare les taux de variation des indices français et américains, on voit que si la chronologie est différente son ampleur a été comparable dans les deux pays (figure 1.15).

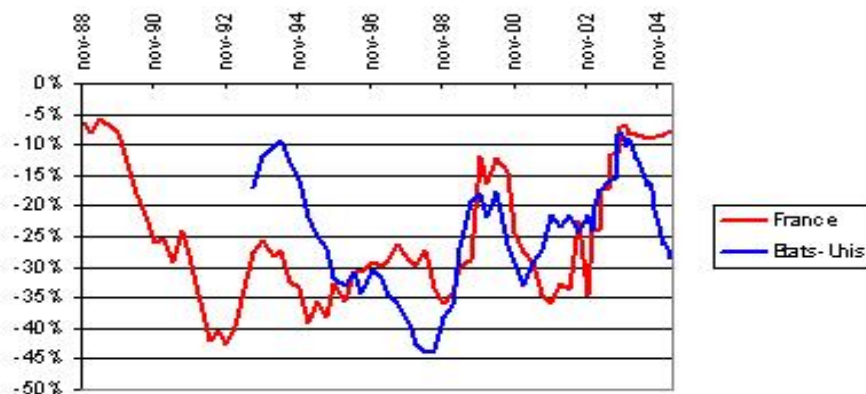


FIG. 1.15 – *Taux de variation des prix français et américain des micro-ordinateurs*

On voit apparaître aux États-Unis une accélération de la baisse à partir de mars 2004 ; ce phénomène ne s'est pas, ou pas encore, produit en France.

1.8 Conjoncture des TIC

Un petit modèle permet d'éclairer la crise que connaissent les TIC (informatique, télécoms) depuis un retournement que l'on peut dater de mars 2000 (effondrement

WPU11510114 pour les « Personal Computers and Workstations », WPU11510115 pour les « Portable Computers »)

du cours de l'action de France Telecom).

Il faut d'abord considérer les lois de la pénétration d'un produit innovant, que « les marchés » (c'est-à-dire la Bourse) ont oublié lorsqu'ils interprétaient les statistiques. Puis, en examinant comment fonctionne la conjoncture des produits arrivés au stade du renouvellement, on peut anticiper les défis auxquels seront confrontés les fournisseurs.

Ce rapide parcours permet de voir que derrière la crise qui a débuté en mars 2000, due à un ralentissement prévisible et pourtant imprévu, se profile une crise plus grave provoquée par un changement de régime conjoncturel auquel les stratégies devront s'adapter.

* *

La pénétration d'un produit nouveau (bien ou service) dans la consommation suit une loi *logistique* : la courbe ressemble d'abord à une exponentielle croissante, puis s'infléchit ; sa croissance ralentit et elle s'aligne enfin sur la pénétration asymptotique. C'est ainsi qu'ont pénétré les produits électroménagers et les services de télécommunications.

La fonction logistique fait partie depuis longtemps de l'arsenal théorique du marketing. L'une de ses expressions est la « loi de Gompertz » :

$$y_t = \delta + \alpha e^{-e^{\beta - \gamma t}}$$

où y_t est le taux de pénétration. Le graphique ci-dessous représente l'évolution de la consommation sur 30 ans en supposant $\alpha = 1$, $\beta = 2,5$, $\gamma = 0,3$, $\delta = 0$. Le niveau asymptotique est $\alpha + \delta$.

Les paramètres ont ici été choisis de sorte que le point d'inflexion, situé au taux de pénétration 36,8 %, se trouve dans l'année 8 (figure 1.16).

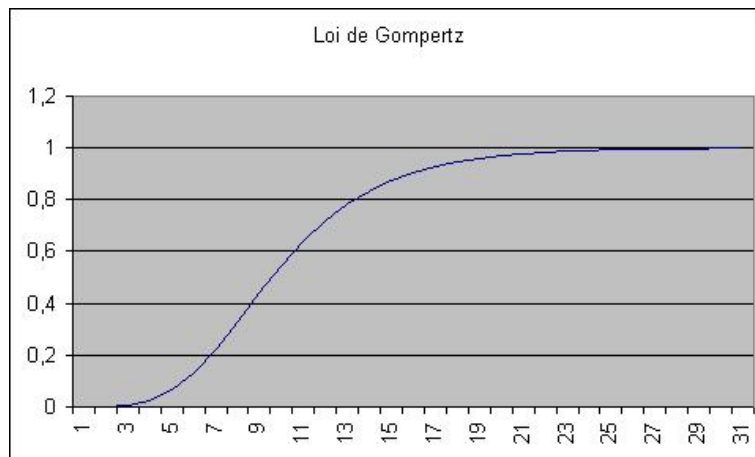


FIG. 1.16 – Loi logistique de pénétration d'un nouveau produit

Lorsqu'un produit est non seulement nouveau, mais innovant, c'est-à-dire lorsqu'il bouscule les usages courants, la logistique est pentue : le démarrage est plus lent (il faut vaincre des réticences), par contre la croissance est plus rapide une fois

qu'elle s'est enclenchée²⁶ (« effet d'avalanche »). L'inflexion est plus lente à venir mais elle se produit tout de même : la prolongation illimitée d'une exponentielle est impossible, même en tenant compte d'un éventuel « multi-équipement » (personnes qui ont plusieurs automobiles, plusieurs téléviseurs, plusieurs ordinateurs, plusieurs téléphones etc.) et de la diversification des services rendus sur la plate-forme technique, une fois celle-ci installée : la consommation de ces services ne peut pas excéder le niveau du PIB !

* *

Les « investisseurs » (c'est ainsi, bizarrement, que l'on appelle les personnes qui achètent des actions à la Bourse) regardent les comptes et les taux de croissance trimestriels. Ils sont moins attentifs aux « fondamentaux » dont fait partie le niveau asymptotique de la pénétration. Ils extrapolent donc la croissance initiale et anticipent l'évolution exponentielle des chiffres d'affaire et des profits. Cette anticipation les incite à attribuer aux entreprises des TIC une valeur élevée. Ainsi valorisées, celles-ci peuvent se procurer des fonds en émettant des actions nouvelles ; leur actif, qui comprend des actions d'autres entreprises du secteur, est lui aussi fortement valorisé. Leur taux d'endettement paraît alors bas ; les banques, alléchées, leur proposent des prêts qui financeront l'achat d'autres entreprises (coup double pour la banque : elle prête à une entreprise solvable et encaisse une commission). Ainsi se met en place un piège.

Ce qui devait se produire finit par arriver : la pénétration s'infléchit, la croissance ralentit. Le dimensionnement des machines et des réseaux, le niveau des stocks d'équipement, préparés pour une demande supérieure, apparaissent alors excessifs (surcapitalisation). Les investisseurs qui anticipaient une exponentielle se trouvent confrontés à une logistique. Leurs anticipations sont modifiées ainsi que leur évaluation des entreprises. Le cours des actions s'effondre. Les actifs se dévalorisent, les bilans sont mécaniquement détériorés. Le taux d'endettement effraie les banques qui, tremblant pour leurs créances, refusent désormais le renouvellement des prêts qui allait auparavant de soi. C'est la faillite. « Ils ne mouroient pas tous, mais tous étoient frappés²⁷ ».

Pourquoi cette épidémie ? *parce que les investisseurs ont pris le début d'une logistique pour une exponentielle.* Ils ont été confortés dans cette erreur par des plus-values substantielles (la croissance exponentielle semblait se confirmer), par les indicateurs de court terme (ils ne montraient au début aucune tendance à l'inflexion), par la conviction immodeste que « les marchés ont toujours raison », par le manque d'attention envers les enseignements classiques du marketing.

Il est vrai qu'un autre phénomène a compliqué l'interprétation de la pénétration : la baisse rapide des prix (35 % par an à qualité constante, 17 % par an en prix moyen pour les micro-ordinateurs) a élargi le marché potentiel et fait monter l'asymptote de la pénétration. La croissance résultait donc de deux mouvements conjugués : une logistique qui se rapprochait de l'asymptote correspondant au prix courant ; une

26. Dans le cas de l'économie des réseaux, on explique l'effet d'avalanche (que l'on nomme alors aussi « effet de réseau ») par le fait que l'utilité d'un raccordement nouveau est fonction croissante du nombre des personnes déjà raccordées. Dans le cas des TIC, l'effet de réseau joue et aussi l'effet de mode (« il faut être sur l'Internet »).

27. Jean de La Fontaine, « Les animaux malades de la peste », *Fables*, Livre VII, Fable I, vers 7.

asymptote qui se déplaçait vers le haut en raison de la baisse du prix.

La liste des difficultés s'arrête là pour les opérateurs de télécommunications dont le revenu est récurrent : le nombre des abonnés peut cesser de croître ainsi que leur consommation, mais ils continuent de payer l'abonnement et de consommer le service. Pour les fournisseurs de composants et d'équipements par contre une autre difficulté se présente : l'évolution apporte un changement de régime, car ils passent d'un marché de premier équipement à un marché de renouvellement dont la conjoncture est beaucoup plus accidentée.

* *

Le marché des téléphones mobiles, des ordinateurs, est l'addition de deux marchés : un marché du premier équipement gouverné par la logistique de pénétration que nous venons d'examiner ; et un marché de renouvellement qui devient de plus en plus important à mesure que la pénétration croît.

Il existe de purs marchés de renouvellement comme celui de l'automobile. Elle a depuis longtemps atteint sa pénétration asymptotique dans les pays riches : les acheteurs de voiture neuve sont pour la plupart d'anciens propriétaires de voiture qui ont voulu acheter un nouveau modèle pour bénéficier d'un accroissement de confort et peut-être de prestige social. La durée de vie de l'automobile est ainsi de l'ordre de cinq à sept ans²⁸.

La durée de vie économique d'une automobile, d'un micro-ordinateur, d'un téléphone mobile, est de quelques années. Après ce délai, ces matériels fonctionnent encore mais ils sont « dépassés » par des innovations survenues entre temps et qui font envie aux utilisateurs : ils sont devenus *obsoletes*.

L'effet de mode, très fort pour les micro-ordinateurs et les téléphones mobiles, rapproche ces produits de l'automobile mais avec une durée de vie économique plus courte, ce qui donne encore plus de poids au phénomène du renouvellement.

L'étude de l'industrie automobile illustre le fait que *le marché des biens durables est un amplificateur de la conjoncture générale*. En effet lorsque la conjoncture est défavorable le remplacement d'une automobile peut être reporté à des jours meilleurs. En revanche lorsqu'elle est favorable les clients se précipitent pour acheter. On représente ces comportements par une probabilité de remplacement croissant d'autant plus vite avec l'âge du véhicule que la conjoncture est meilleure. Le renouvellement est une fonction de la pyramide des âges et de la conjoncture ; par ailleurs, il modifie la pyramide des âges. La rétroaction que comporte cette dynamique amplifie les fluctuations de la conjoncture.

On peut en tirer une leçon : dès que la part du renouvellement dans le marché des TIC sera importante, *la conjoncture du marché des composants et des équipements sera un amplificateur de la conjoncture générale*. Cela posera aux entreprises des TIC un défi auquel rien ne les a préparées.

* *

28. Il n'en est pas de même du marché des réfrigérateurs et des machines à laver, qui engagent moins le prestige social : on ne remplace ces machines que lorsqu'elles sont en panne et leur durée de vie est de l'ordre de vingt ans. Le marché du premier équipement (équipement des logements neufs) reste donc significatif pour ces produits.

On peut dès lors découper l'évolution du marché des TIC en trois phases :

1) *croissance exponentielle* : le marché est dans une logique de pénétration et de premier équipement, soutenue par la baisse tendancielle du prix ; la croissance semble exponentielle. Les entreprises, conquérantes et optimistes, investissent massivement pour tirer parti de l'élargissement tendanciel du marché. Les investisseurs extrapolent la croissance et attribuent une forte valeur aux entreprises.

2) *ralentissement* : deux logiques se superposent : la pénétration ralentit, et le marché de renouvellement, qui représente une part croissante du débouché, est soumis à de fortes fluctuations conjoncturelles. Les entreprises, surprises par le ralentissement, doivent réviser leur stratégie de croissance. Certaines se trouvent suréquipées. Les fluctuations les surprennent également. Les investisseurs ajustent leurs perspectives à la baisse et sont déconcertés par les fluctuations des chiffres d'affaires : le cours des actions chute, des faillites se produisent (Worldcom), les fournisseurs de composants et d'équipements s'inquiètent et s'activent pour trouver de nouveaux débouchés (Intel, Alcatel, Northern, Ericsson etc.) C'est la situation actuelle.

3) *maturité* : la pénétration a atteint son niveau asymptotique, la croissance suit celle de l'économie, le marché est principalement un marché de renouvellement. Les fournisseurs de composants et d'équipements sont soumis aux mêmes aléas conjoncturels que ceux que connaît aujourd'hui l'industrie automobile. Ils doivent doser leurs investissements, la gestion de leurs stocks et le lancement des nouveaux modèles en fonction de la conjoncture. Le cours des actions se compose, selon la proportion des spéculateurs parmi les investisseurs, de l'addition d'une part liée aux fondamentaux et d'une part liée à la conjoncture. La physionomie du secteur a profondément changé.

1.8.1 Simulation

Pour illustrer le scénario ci-dessus, nous construirons un petit modèle. Nous négligerons l'effet de la baisse tendancielle du prix des TIC : elle fait monter l'asymptote de la pénétration, mais cette hausse devient de plus en plus faible lorsque la baisse du prix se poursuit. Supposons que la croissance tendancielle du PIB soit de 3 % par an et qu'il subisse des oscillations conjoncturelles d'une périodicité de cinq ans. Le PIB de l'année t est donné par l'expression

$$PIB_t = PIB_0[1 + a.\sin(\omega t)]e^{rt}$$

avec $r = 0,03$, $a = 0,02$ et $\omega = 2\pi/5$. Le graphique 1.17 couvre 30 ans.

Supposons que la pénétration suive la loi de Gompertz évoquée ci-dessus et que le niveau asymptotique de cette loi évolue comme le PIB ; le parc installé lors de l'année t est obtenu en combinant les deux relations précédentes :

$$Parc_t = y_t PIB_t$$

On remarque sur le graphique 1.18 que l'influence des oscillations conjoncturelles sur le niveau du parc est peu sensible dans la phase de pénétration initiale et devient de plus en plus sensible par la suite.

Le renouvellement du parc se fait de façon différente selon que l'économie se trouve en haut ou en bas du cycle conjoncturel. Nous supposerons que les probabilités

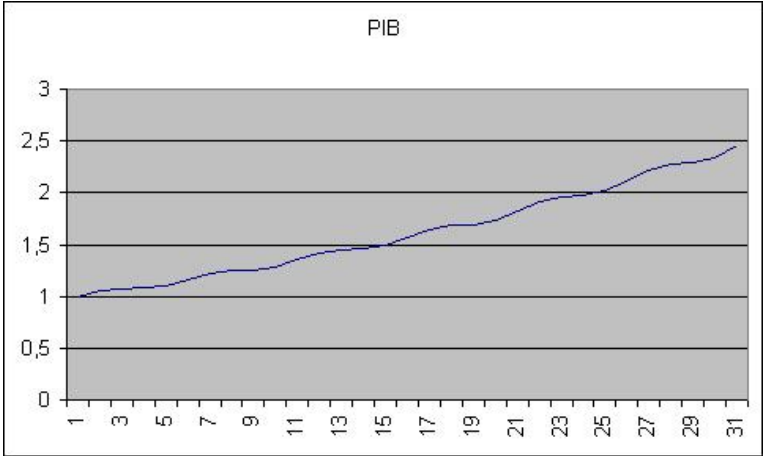


FIG. 1.17 – Croissance tendancielle et fluctuations conjoncturelles

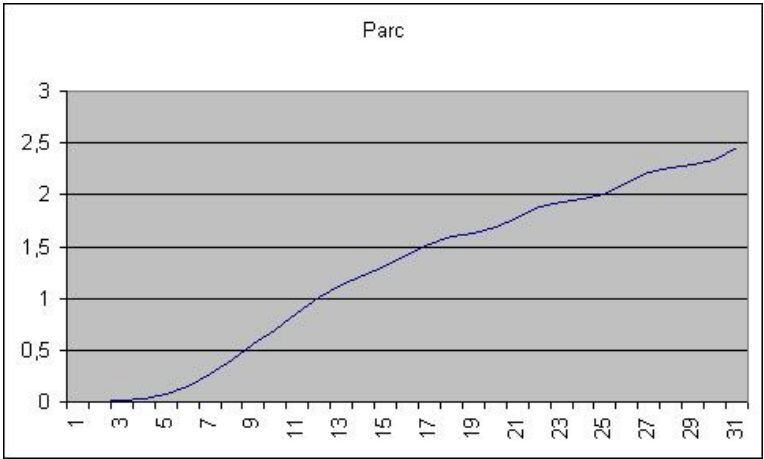


FIG. 1.18 – Evolution du parc installé

P_A de renouvellement du matériel en fonction de l'âge sont celles données par le tableau 1.1.

Les distributions des durées de vie qui correspondent à ces probabilités sont représentées par le graphique 1.19, la probabilité d'une durée de vie D étant :

$$Prob_D = P_D \prod_{A=0}^{D-1} (1 - P_A)$$

Si la conjoncture restait durablement en haut de cycle, la durée de vie moyenne $\sum_{D=0}^{\infty} D Prob_D$ serait de 2,78 ans ; en bas de cycle, elle serait de 3,89 ans.

Nous supposons que dans les situations intermédiaires la probabilité de renouvel-

Année	0	1	2	3	4	5	6	7
Haut de cycle	0	0,2	0,4	0,7	0,9	1	1	1
Bas de cycle	0	0,1	0,2	0,35	0,45	0,5	0,6	0,7

TAB. 1.1 – Probabilité de renouvellement en fonction de l'âge du matériel

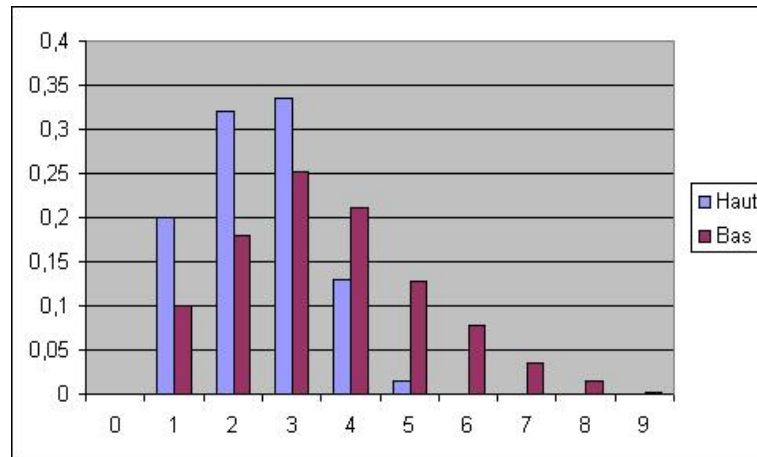


FIG. 1.19 – Distributions des durées de vie

lement est une moyenne pondérée des deux valeurs extrêmes. Notons $\varphi(t)$ la fonction qui indique à quel point du cycle se trouve l'économie :

$$\varphi(t) = [1 + \sin(\omega t)]/2$$

La probabilité du renouvellement d'un ordinateur d'âge A lors de l'année t est alors, en notant respectivement P_{AH} et P_{AB} les probabilités pour le renouvellement d'un ordinateur d'âge A en haut et en bas de cycle :

$$P_{A,t} = \varphi(t)P_{AH} + [1 - \varphi(t)]P_{AB}$$

La production lors de l'année t doit permettre l'accroissement annuel du parc et le renouvellement des ordinateurs obsolètes soit, en notant $Parc_{A,t}$ l'effectif du parc d'âge A lors de l'année t :

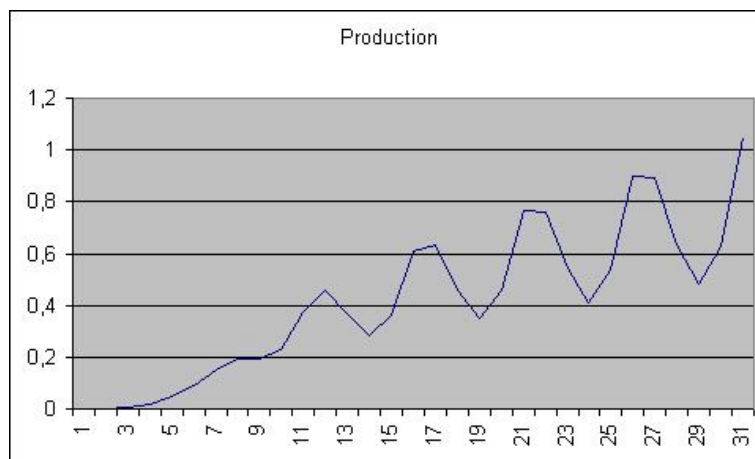
$$Prod_t = Parc_t - Parc_{t-1} + \sum_A P_{A,t} Parc_{A-1,t-1}$$

L'évolution de la production a alors l'allure qu'indique le graphique 1.20.

Ce graphique un peu effrayant fait apparaître, lorsque la pénétration est assez forte pour que le renouvellement soit significatif, une succession de hausses et de baisses très importantes de la production. Il illustre l'amplification de la conjoncture que suscite l'alternance des échéanciers de renouvellement : la production est *beaucoup plus accidentée* que la croissance du parc.

1.8.2 Conséquences

Les entreprises du secteur des TIC sont en train de subir un ralentissement qui révèle un surdimensionnement, le secteur s'étant préparé à la poursuite indéfinie d'une croissance exponentielle. Il s'ensuit une dévalorisation des actifs physiques (et pas seulement des actifs boursiers), un effondrement des commandes passées aux fournisseurs et équipementiers (ce qui suscite chez eux une crise encore plus grave), bref un « coup de frein » dont les conséquences s'amplifient chez les fournisseurs de la même façon qu'un ralentissement sur une autoroute se traduit en amont par un bouchon où les véhicules sont à l'arrêt.

FIG. 1.20 – *Évolution de la production*

La dévalorisation des actifs va provoquer des disparitions d'entreprises jusqu'à ce que le niveau des actifs redevienne compatible avec les perspectives de croissance révisées.

Après ce choc, qui résulte de la révision des anticipations et de la prise de conscience que la pénétration est non pas exponentielle mais logistique, se profile pour les fournisseurs de composants et d'équipements un second choc encore plus violent : l'adaptation à un marché de renouvellement.

Les entreprises des TIC, conquérantes et orgueilleuses (d'autant plus que leurs dirigeants, ayant été les premiers à se placer sur un marché à fort potentiel, ont bénéficié d'une réputation de « génie »), vont devoir devenir modestes, adopter les stratégies prudentes de l'industrie automobile, pratiquer un marketing attentif à des fluctuations de la demande qu'il leur faudra savoir anticiper.

Cette adaptation sera culturellement difficile pour les dirigeants du secteur des TIC. Elle risque de susciter une nouvelle vague de faillites due cette fois non à la dévalorisation des actifs, mais à la dévalorisation de la stratégie.

Chapitre 2

Automatisme et intelligence

« Ne tournons pas nos regards vers l'Amérique pour copier servilement les institutions qu'elle s'est données, mais pour mieux comprendre celles qui nous conviennent, moins pour y puiser des exemples que des enseignements, pour lui emprunter les principes plutôt que les détails de ses lois. » (Alexis de Tocqueville, avertissement de la douzième édition de *De la démocratie en Amérique*, [19] p. XLIV).

L'histoire de l'informatique s'éclaire si on la situe dans la perspective de l'industrialisation du travail de bureau, dont on peut situer le début vers 1880 à Chicago. Longtemps utilisée pour accroître la productivité du travail en automatisant les tâches répétitives, l'informatique s'est enrichie dans les années 1990 des apports de la bureautique communicante. Elle équipe désormais les processus de production au plus près de leur déroulement. L'entreprise, entrée dans l'ère du *travail assisté par ordinateur*, doit alors s'efforcer d'articuler au mieux deux ressources : l'organisation de l'être humain et l'automate programmable. Cette tâche n'est pas aisée car l'ordre n'est pas l'état naturel d'un système d'information, sa qualité étant sapée par une entropie croissante.

* *

Même si la contribution des Européens à l'informatique a été importante, ses progrès essentiels ont tous été réalisés aux États-Unis. L'informatique étant à la fine pointe de l'organisation de l'entreprise, elle est toute naturelle pour des Américains dont le pays a, dès sa formation, attribué à l'Entreprise l'hégémonie culturelle qui en Europe appartient à l'État (Gramsci [40] p. 476).

Pour réussir l'articulation du travail humain et de l'automate, il faut percevoir la différence entre l'ordinateur et le cerveau humain : on ne peut pas en effet articuler deux êtres dont on aurait postulé l'identité. Cette question a été obscurcie par les controverses passionnelles sur l'« intelligence de l'ordinateur », controverses qui s'éclairent si on les situe sur l'arrière-plan culturel des États-Unis : est-il possible de planifier l'action au point que l'être humain, pour pouvoir être efficace, doive se faire l'exécutant d'un automate pré-programmé ? Ou bien l'action, avec ses incertitudes, nécessite-t-elle que l'on préserve la capacité de synthèse, de compréhension et de décision propre au cerveau humain ?

2.1 Informatisation de l'entreprise

Pour décrire l'évolution historique du rôle de l'informatique, nous allons d'abord présenter de façon schématisée comment une entreprise travaille, puis montrer comment l'informatique a progressivement équipé les divers types de fonctions assumées au sein de l'entreprise.

Toute activité d'une entreprise débute par des événements externes à la production (ou « événements externes » tout court) : commandes des clients, livraisons des fournisseurs. Lorsqu'il est initialisé par une commande d'un client, le cycle de l'activité se boucle par une livraison accompagnée d'une facturation qui sera suivie d'un paiement (figure 2.1).

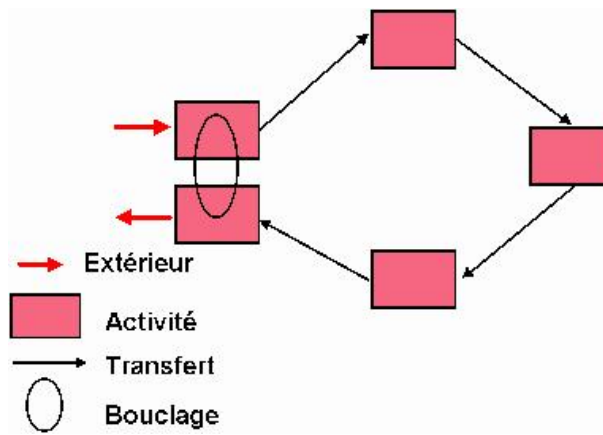


FIG. 2.1 – *Bouclage du processus de production*

Le schéma ci-dessus s'applique à toute entreprise, qu'elle soit informatisée ou non : des « sous-processus » font progresser le processus de production en lui fournissant des « livrables », produits intermédiaires documentaires ou physiques dont la mise à disposition est un « événement interne » (figure 2.2).

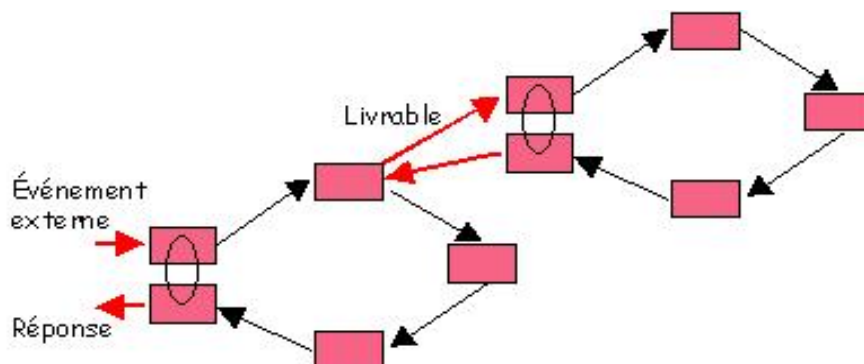


FIG. 2.2 – *Processus et sous-processus*

2.1.1 Organisation du travail de bureau : années 1880

L'organisation du travail de bureau a résulté d'un effort prolongé et méthodique. Les progrès essentiels ont été réalisés lors des dernières décennies du XIX^e siècle dans le « loop » de Chicago, centre d'affaires en croissance rapide où furent mises au point les méthodes de standardisation et de classement documentaire ainsi que l'architecture des grands immeubles de bureau. C'est également aux États-Unis que débuta la mécanisation du travail de bureau avec les machines à écrire et les machines à calculer : la machine à écrire a permis d'obtenir des documents plus lisibles que les manuscrits et de les dupliquer en quelques exemplaires grâce au papier carbone puis au stencil ; la machine à calculer a facilité les opérations de vérification et de calcul.

Les tâches de bureau remplies par les employés dans la première moitié du XX^e siècle se classent en deux catégories : celles internes à l'entreprise (« middle office » et « back office »), qui peuvent être entièrement organisées ; celles effectuées au contact des clients ou des fournisseurs, qui impliquent une part de dialogue et de négociation (« première ligne » ou « front office »).

* *

Les tâches du « back office » obéissaient toutes à un même schéma :

1) d'une part l'agent reçoit des commandes, des matières premières ou des produits intermédiaires ; son travail consiste à élaborer d'autres produits intermédiaires (« livrables ») qu'il oriente vers l'étape suivante du processus. Par exemple, pour les agents qui traitaient l'information dans les grandes banques ou compagnies d'assurance, le travail se faisait sur un bureau dans une salle où se trouvaient de nombreux employés ; à gauche de l'agent se trouvait la barquette arrivée, à droite la barquette départ, les dossiers étant apportés et emportés par des personnes équipées de caddies (figure 2.3).

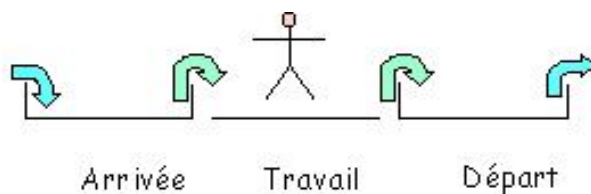


FIG. 2.3 – Poste de travail du « back-office »

2) d'autre part les personnes qui transportent les dossiers d'un bureau à l'autre, ainsi que le superviseur de la salle de travail, assurent une logistique (« middle office ») qui entoure les tâches des agents d'un réseau de communication et de contrôle. La mesure du flux quotidien permet d'établir des normes de productivité. Les délais normaux de traitement d'une affaire peuvent être évalués. La pile de dossiers qui reste dans la barquette arrivée signale un agent qui travaille plus lentement que les autres.

Le travail que l'agent effectue sur un dossier consiste en calculs, vérifications et transcriptions, et aussi en expertises, classements, évaluations et décisions (ou

formulation d'avis pour préparer les décisions). En même temps qu'il fait progresser le processus de traitement des affaires, ce travail alimente des fichiers manuels qui constituent la mémoire de masse de l'entreprise. Les éventuelles interrogations donnent occasion à des échanges de notes ou de fiches que l'agent place dans la barquette « départ » en mentionnant le nom du destinataire, les réponses parvenant dans la barquette « arrivée » avec les dossiers à traiter.

Évolution des équipements de bureau

Les équipements du travail de bureau (fauteuils, bureaux, téléphones, photocopieurs, télécopieurs, calculateurs, machines à écrire, classeurs, trombones, post-its, sans même évoquer l'ordinateur et sa séquelle d'imprimantes, scanners etc.) sont tous d'origine récente : le brevet du trombone est déposé en 1901, celui du classeur mécanique en 1904. Les copieurs apparaissent en 1890, mais la photocopie ne se répandra vraiment qu'à partir de 1960 avec la xérogaphie. Le Post-it^a sera lancé par 3M (après de longues hésitations) en 1980.

La machine à écrire (D. Rehr [71]), inventée en 1868 par l'Américain Christopher Latham Sholes, est commercialisée par Remington en 1874. Elle a déjà le clavier QWERTY mais elle écrit en majuscules et l'auteur ne peut pas voir le texte qu'il tape. 5 000 machines sont vendues en cinq ans. La Remington n° 2 de 1878 permet d'écrire en minuscules et majuscules. En 1895, Underwood commercialise une machine qui permet de voir ce que l'on tape. Dès lors la machine à écrire se répand rapidement dans les entreprises. La première école de dactylographie est créée en 1911.

^a www.3m.com/about3M/pioneers/fry.html

Dans les entreprises industrielles, le travail de bureau traitait les commandes, les factures et la comptabilité ; il émettait les ordres qui déclenchaient les opérations physiques de production, approvisionnement, stockage, transport et livraison. Les décisions concernant les opérations physiques étaient prises dans les bureaux, les décisions laissées aux agents de terrain étant celles qui accompagnent l'exécution de ces opérations.

* *

Processus de l'entreprise industrielle

Dans l'entreprise industrielle, les commandes sont satisfaites en puisant dans les stocks ; le suivi statistique du flux de commandes permet d'évaluer la demande anticipée et de déterminer le programme de production ; les facteurs de production (capital K , travail L , biens intermédiaires X) sont mobilisés chacun selon le cycle de vie qui lui est propre ; la fonction de production $Y = f(K, L, X)$ est mise en œuvre pour alimenter les stocks (figure 2.4).

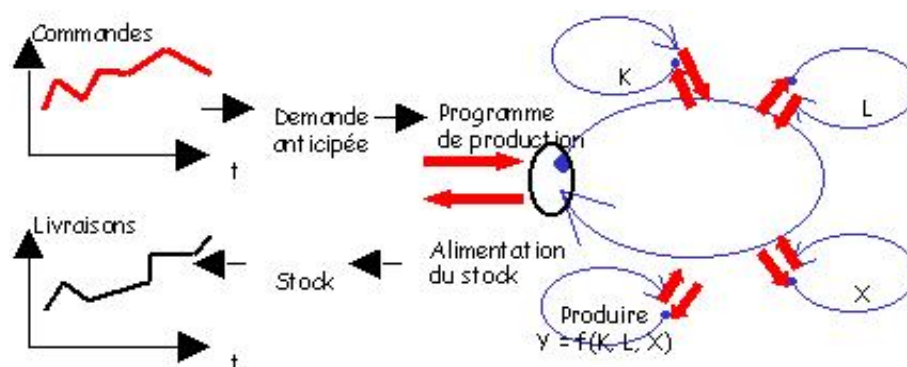


FIG. 2.4 – Anticipation et stockage dans l'entreprise industrielle

La procédure du « front office » était plus souple (figure 2.5), car il travaillait au contact d'un client ou d'un fournisseur, que ce soit par contact « présentiel », par téléphone ou par courrier : il ne s'agissait plus de traiter des documents conformes aux formats types de l'entreprise mais de répondre à des demandes ou questions formulées dans la langue de personnes extérieures à l'entreprise et dans un ordre correspondant à leurs priorités.

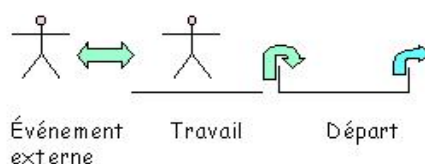


FIG. 2.5 – Poste de travail du « front-office »

L'agent du « front office » devait transcrire les indications recueillies lors de la relation externe en un document susceptible d'alimenter le processus interne.

* *

Cette organisation comportait des articulations fragiles. Les documents posés en pile risquaient d'être traités sur le mode LIFO (« last in, first out ») qui induit des délais aléatoires ; la succession des transferts entre agents pouvait finir « dans les sables » en cas d'erreur d'aiguillage ; si l'on avait besoin de retrouver un dossier en cours de traitement, il n'était pas facile de le suivre à la trace le long de son parcours. Enfin, le schéma que nous avons décrit se dégradait en variantes artisanales dans les entreprises petites et moyennes et il restait vulnérable à la négligence ou à l'étourderie.

2.1.2 Arrivée de l'informatique : années 1950

L'industrialisation du travail de bureau, avec les armoires de dossiers suspendus, classeurs, bibliothèques tournantes, la logistique du transport des dossiers, les longues opérations de calcul, appelait l'informatique. Mais l'informatisation n'a pris

son essor que dans les années 1950, la guerre ayant pendant dix ans bloqué l'utilisation civile des techniques tout en accélérant leur conception (comme ce fut le cas pour l'agriculture : en Europe le tracteur ne se répandra pas avant les années 1950).

La mécanographie, fondée sur le traitement électromécanique de cartes perforées par des trieuses et tabulatrices, a été d'abord conçue pour réaliser des travaux statistiques. La première réalisation est celle du statisticien américain Herman Hollerith (1860-1829) pour le recensement de la population des États-Unis en 1890. Les entreprises créées par Hollerith sont à l'origine d'IBM¹. Les premiers utilisateurs de la mécanographie furent les instituts statistiques, les armées et quelques administrations (Carmille [13]). Les origines de plusieurs grands groupes informatiques remontent à l'ère de la mécanographie².

C'est avec l'ordinateur, plus puissant que la machine mécanographique et surtout plus souple grâce à la mise en œuvre automatique de programmes enregistrés conformément à l'architecture de von Neumann (von Neumann [88]) que l'informatique a pénétré les entreprises dans les années 1950 et surtout 1960. Elle a été d'abord utilisée pour automatiser la production physique : dès 1968, on a pensé à remplacer la commande numérique des machine-outils par la « commande numérique directe ». Dans le numéro spécial de « Science et Vie » sur l'automatisme en 1964, la gestion n'apparaît encore que comme un domaine relativement secondaire pour l'automatisation.

Partage du travail entre l'ordinateur et l'être humain

Les entreprises achètent les ordinateurs pour économiser le temps que les agents opérationnels passent à des opérations répétitives de vérification, calcul et transcription, et aussi pour obtenir plus rapidement des informations de gestion d'une meilleure qualité.

Elles utilisent la machine pour faire des traitements (puissance) ainsi que pour classer et trier les données (mémoire).

Elles réservent à l'être humain les fonctions où il est supérieur à l'ordinateur : comprendre, expliquer, décider, concevoir.

Les premières entreprises de service à s'informatiser furent les banques et assu-

1. Voir Donald E. Knuth [51] volume 3 p. 383. Hollerith fonde en 1896 la « Tabulating Machine Company ». Cette compagnie fusionne en 1911 avec la « Computing Scale Company of America » (balances automatiques) et l'« International Time Recording Company » (horloges enregistreuses) pour former la « Computing Tabulating Recording Company » (CTR) dont la direction est confiée à Thomas J. Watson (1874-1956). Ce dernier, devinant le potentiel de la mécanographie pour la gestion, change en 1917 le nom de la filiale canadienne de la CTR en « International Business Machines » (IBM). La CTR adopte elle-même le nom d'IBM en 1924.

2. Le bureau du Census demanda à James Powers de fabriquer les machines pour le recensement de 1910. Powers fonda en 1911 l'« Accounting and Tabulating Machine Company » : Elle fusionna avec la « Remington Typewriter Company », présidée par James Rand, pour devenir en 1927 la « Remington Rand » qui fusionnera en 1955 avec « Sperry Gyroscope » pour former « Sperry Rand ». Frederick Bull (1882-1925), ingénieur norvégien, créa en 1922 avec son associé Kurt Kruesen une société de fabrication de machines mécanographiques. Les brevets furent achetés en 1929 par le groupe suisse H. W. Egli qui fonda la compagnie « Egli-Bull ». En 1932 le groupe français Caillies racheta les brevets au groupe « Egli-Bull » pour créer en 1933 la « Société des Machines Bull ».

rances ; dans les autres secteurs, les premières utilisations ont concerné la comptabilité, la paie et la gestion des stocks. Cela a modifié les conditions *physiques* du travail : les employés passaient dans les années 60 une partie de leur temps à perforer des cartes et dépouiller des « listings » ; puis dans les années 70 et 80 on installa des terminaux qui seront dans les années 90 remplacés par des micro-ordinateurs en réseau. À chaque étape, l'ergonomie s'est modifiée ainsi que les possibilités offertes à l'utilisateur.

Décalage de la pénétration des innovations

Lorsqu'on examine comment l'informatique a pénétré les entreprises, on constate un décalage temporel important entre la disponibilité d'une innovation et sa mise en œuvre. La chronologie des innovations est donc différente de celle de leur utilisation par les entreprises.

Ainsi, il était dès 1957 possible d'utiliser quatre terminaux en grappe sur l'IBM 305 ; mais les entreprises en sont restées pendant les années 60 au couple « carte perforée et listing » et la diffusion des terminaux date des années 70. De même, il était dès le début des années 80 possible de fournir aux utilisateurs des micro-ordinateurs en réseau ; mais de nombreuses entreprises ont continué à utiliser des terminaux « passifs » jusqu'au milieu des années 90.

Ces décalages s'expliquent : la première version d'une solution innovante est coûteuse et demande des mises au point, sa mise en œuvre implique des changements également coûteux de l'organisation. L'entreprise prendra donc tout son temps avant de comprendre l'utilité d'une innovation, puis de la mettre en œuvre.

L'espace de travail change alors d'allure. Même si le « bureau sans papier » reste rare, les archives et dossiers sur papier sont remplacés, dans une large mesure, par des informations stockées dans les mémoires électroniques. L'interface avec écran, clavier et souris s'installe sur tous les bureaux. Une part croissante du travail à faire arrive non plus dans une barquette, mais sur l'écran *via* le réseau.

Ce changement ne modifie pas fondamentalement la nature du travail : la différence entre événement interne et événement externe reste de même nature, même si l'écran-clavier s'impose désormais comme un tiers dans la relation avec les personnes extérieures à l'entreprise - au point parfois de gêner le dialogue (figure 2.6).

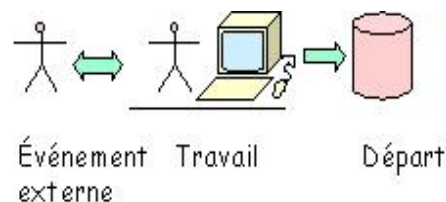


FIG. 2.6 – L'informatisation du « front-office »

Toutefois l'agent n'a plus, en principe, à recopier une information déjà introduite dans l'ordinateur ; la vérification de la saisie peut être automatique ; les calculs (de

comptes, prix, taxes, salaires, ainsi que les totalisations etc.) sont eux aussi automatisés, ainsi que la mise en forme et l'édition des divers « états » (bulletins de paie, documents comptables, état des stocks, statistiques etc.)

L'ordinateur remplit ainsi deux fonctions : d'une part il aide à traiter des dossiers individuels dont il facilite aussi le tri et la recherche ; d'autre part il permet de produire des indicateurs. L'être humain se spécialise dans les tâches qu'il fait mieux que l'ordinateur : il analyse l'information pour faire le tour d'un problème, l'interprète pour comprendre, la synthétise pour résumer et communiquer ce qu'il a compris ; enfin il décide ou même il conçoit. Ayant été soulagé des travaux qui exigeaient une utilisation répétitive de son cerveau, il est invité à se consacrer à des travaux auxquels cet organe est le mieux adapté. On arrive ainsi à un partage des tâches où chacune des deux ressources (le « silicium », la « matière grise ») tend à être utilisée au mieux de ses aptitudes. Toutefois cette évolution n'est pas facile.

Une évolution difficile

L'évolution est pénible pour ceux des agents, parfois les plus intelligents, qui avaient pris l'habitude de travailler de façon mécanique et rapide tout en pensant à autre chose. Désormais le travail leur demande concentration, réflexion, responsabilité, prise de risque. Même s'il est en principe devenu plus intéressant, il implique un effort psychologique accru.

Il faut aussi des changements dans l'organisation (transversalité etc.) : l'entreprise, qui doit accorder à l'employé un pouvoir de décision correspondant aux responsabilités qu'elle lui confie, doit aussi éviter de le harceler pour obtenir toujours plus de productivité, de qualité ou de profit unitaire. Les rapports entre personnes doivent devenir *respectueux* : dans une entreprise où la prise de décision est décentralisée, il faut en effet savoir *écouter ce que dit l'autre*.

Les entreprises sont parfois tentées d'oublier que l'on ne peut pas demander à un salarié d'être à la fois un exécutant docile et un pionnier plein d'initiative et de créativité.

Nous sommes là vers le milieu des années 1980. Il faut compléter cette description en mentionnant des défauts souvent rencontrés. D'une part les « applications » informatiques ont été conçues séparément et communiquent mal : les agents doivent dans le cours d'une même tâche ouvrir une session, puis la fermer pour passer à une autre dont l'ergonomie sera différente, ressaisir des données, utiliser des codes divers dont la mémorisation demande un apprentissage. Si l'informatique est puissante, elle manque encore de cohérence et de « convivialité ».

C'est que l'automate n'est pas souple - et comme il ne s'adapte pas facilement aux utilisateurs, l'entreprise leur demande de s'adapter à lui. Ses défauts sont d'abord tolérés en raison des gains d'efficacité que l'informatique apporte. Mais ils deviennent de plus en plus insupportables.

Le « système d'information » vise à les corriger. Les diverses applications doivent s'appuyer sur un référentiel unique, ce qui garantit leur cohérence sémantique ; elles doivent échanger les données et se tenir à jour mutuellement, ce qui assure la cohérence de leur contenu et supprime les ressaisies.

Toutefois cette mise en ordre reste souvent partielle et les défauts persistent en raison du poids de l'existant et de la pression d'autres priorités.

Écart entre théorie et pratique

L'idée du système d'information n'est pas nouvelle : la théorie était déjà bien avancée avant la seconde guerre mondiale. Mais il faut, quand on examine la pratique des entreprises, tenir compte de l'écart chronologique entre l'émission d'une idée et sa mise en œuvre. La lenteur du cycle de vie de l'organisation fait que des méthodes que chacun sait absurdes survivent alors que la mise en œuvre de solutions simples et connues est ajournée.

2.1.3 La bureautique communicante : à partir des années 1980

L'arrivée du micro-ordinateur dans les années 80 (Penny et Volle, [33]) fut un choc pour les informaticiens qui ne reconnurent pas immédiatement sa légitimité ni son utilité. Le micro-ordinateur servit d'abord à diffuser les applications de bureautique personnelle qui avaient été mises au point sur des architectures de mini-ordinateurs en grappe (traitement de texte, tableur, grapheur). Il supplanta progressivement la machine à écrire et la machine à calculer. Cependant les applications bureautiques se sont déployées dans le désordre (versions différentes des applications, travaux locaux sans cohérence d'ensemble).

Au début des années 90 la mise en réseau des micro-ordinateurs a confronté la bureautique aux exigences de cohérence du système d'information : pour toute donnée importante, seule doit exister sur le réseau une mesure définie et tenue à jour par le propriétaire de la donnée.

Le micro-ordinateur a enfin cumulé divers rôles : d'une part il remplace les terminaux pour l'accès aux applications centrales, d'autre part il apporte à l'utilisateur la bureautique personnelle, puis la « bureautique communicante » (messagerie, documentation électronique, « groupware » puis Intranet). Le PC en réseau devient ainsi tout à la fois le terminal ergonomique des applications centrales, un outil de communication asynchrone entre agents et la porte d'accès aux ressources documentaires de l'entreprise (figure 2.7).

* *

On dirait alors que l'informatique a accompli tout ce qui lui était possible : elle fournit à l'utilisateur une interface qui, fédérant sous une ergonomie cohérente les accès aux diverses applications, lui évite les connexions-déconnexions et les doubles saisies tout en soulageant son effort de mémoire ; elle fournit aussi un média de communication. Il lui reste cependant à assister les utilisateurs non seulement dans chacune de leurs tâches considérée séparément, mais dans la succession et l'articulation des activités tout au long du processus de production.

En effet si l'informatique a libéré l'agent des tâches répétitives de calcul, vérification et transcription, les entreprises ne l'ont pas encore pleinement utilisée pour

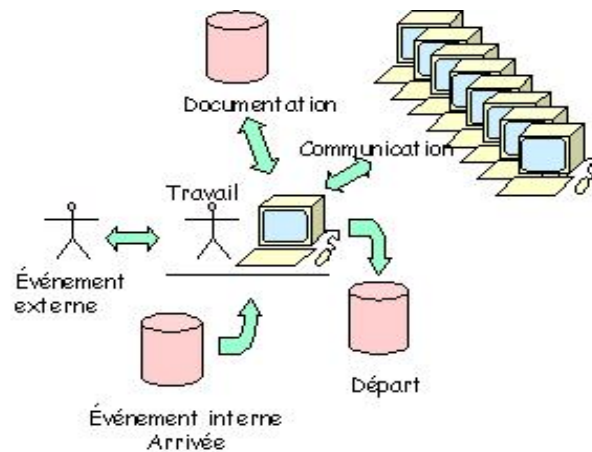


FIG. 2.7 – Poste de travail informatisé

assurer les fonctions de logistique et de supervision, de « middle office », remplies autrefois par les personnes qui transportaient les dossiers et par les superviseurs des salles de travail. Or le travail, devenu informatique (« virtuel »), a perdu la visibilité que lui conférait l'apparence physique des documents et dossiers sur papier. Il est devenu plus difficile de vérifier sa qualité, d'évaluer la productivité des agents et de maîtriser les délais de production.

Rien de tout cela n'est cependant impossible pour l'informatique. Les outils existent depuis longtemps (les premiers « workflows » ont fonctionné dès l'époque des *mainframes*), mais pour qu'ils soient mis en œuvre il faut que le besoin soit ressenti et que la possibilité de le satisfaire soit perçue. L'attention s'était d'abord focalisée sur la productivité de l'agent individuel ainsi que sur la maîtrise des *concepts* (composants, classes, attributs, fonctions) que l'informatique mettait à sa disposition. Il fallait maintenant utiliser celle-ci pour automatiser les *processus* de production eux-mêmes - ou, plus précisément, pour apporter l'assistance de l'automate aux agents opérationnels chargés de ces processus.

L'informatique communicante apporte un élément de solution : s'il est possible aux utilisateurs de communiquer par messagerie, pourquoi ne pas utiliser ce média pour tisser une solidarité entre les étapes d'un même processus?

2.1.4 Du concept au processus : années 1990

Pour retrouver la maîtrise de la logistique que l'informatique avait dans un premier temps négligée, il fallait introduire dans le système d'information les tables d'adressage qui balisent les transferts entre agents successifs, la traçabilité (possibilité de retrouver et consulter un dossier en cours de traitement), des indicateurs de volume, de délai et de qualité : ce sont là les fonctionnalités du *workflow*³. Celui-ci améliore d'ailleurs notablement la logistique par rapport à l'époque du papier : il supprime le risque du « last in, first out », assure la traçabilité des dossiers et produit automatiquement des indicateurs de volume et de délai qui facilitent la maîtrise de la qualité.

3. Le terme « workflow » désigne l'informatisation d'un processus.

Dès lors, le système d'information équipe les processus internes de l'entreprise au plus près de la pratique professionnelle en articulant, selon une frontière d'ailleurs délicate, les fonctionnalités de l'informatique de communication à celles du traitement des données structurées (figure 2.8).

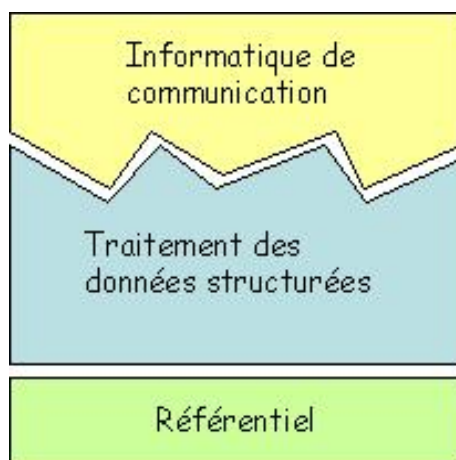


FIG. 2.8 – *Les trois couches du système d'information*

Pour concevoir le traitement des données structurées, il avait fallu concentrer l'attention sur les concepts à l'œuvre dans le système d'information et sur le processus des traitements informatiques. Pour concevoir un workflow, il faut concentrer l'attention sur l'enchaînement des tâches des agents et donc sur le processus opérationnel. Celui-ci se complique d'ailleurs avec l'arrivée du multimédia pour les événements externes (utilisation conjointe du courrier, du téléphone, du face-à-face, de l'Internet, de la carte à puce) comme pour les événements internes (Intranet etc.), et aussi avec l'interopérabilité des systèmes d'information que nécessitent les partenariats. La hiérarchie des difficultés invite à examiner en priorité le processus opérationnel : cet examen dictera les concepts et la dynamique sur lesquels se fonde le traitement des données.

Alors qu'auparavant la pratique professionnelle avait été invitée à se construire autour de l'informatique, désormais l'informatique se construit autour de la pratique professionnelle. Ce changement de point de vue s'accompagne, en ce qui concerne l'organisation, de l'émergence dans les métiers de l'entreprise de spécialistes de la *maîtrise d'ouvrage* du système d'information. Pour prendre en compte de façon exacte le déroulement des processus au sein des métiers, il faut en effet à la fois une proximité quotidienne avec les agents opérationnels et une rigueur intellectuelle dont le besoin n'avait pas jusqu'alors été ressenti. Ces professionnels mettent en forme les processus opérationnels en utilisant par exemple le langage UML⁴.

De nouveaux problèmes apparaissent alors : comment choisir, si l'on veut un système d'information assez sobre pour pouvoir évoluer, entre les fonctionnalités que l'on fournira et celles sur lesquelles on fera l'impasse ? Comment faire en sorte que le métier, ses dirigeants, s'approprient le système d'information, valident ses spécifications, participent à sa définition ? Par ailleurs, si la maîtrise du processus convient

4. « Unified Modeling Language » (Booch, Rumbaugh et Jacobson [28]). On peut aussi consulter www.rational.com.

aux travaux internes, il sera beaucoup plus difficile d'outiller l'agent du « front office », qui travaille au contact des clients ou des fournisseurs : on ne peut pas prévoir en effet l'ordre dans lequel il devra saisir les données et lancer les traitements. Tout au plus le système d'information peut-il lui fournir une aide contextuelle et la liste des tâches à accomplir, équipée de boutons indiquant pour chaque tâche le degré d'avancement ; le workflow ne débute qu'au moment où l'agent alimente les événements internes.

* *

Le resserrement des relations entre l'informatique communicante et le traitement des données structurées amène à construire un système d'information « sur mesures », « près du corps », dont la définition et l'évolution adhèrent à la pratique professionnelle des agents. Il permet d'associer aux données leur commentaire, ce qui les rend compréhensibles, facilite leur transformation en information et modifie donc d'autant leur rôle dans l'entreprise⁵.

Le système d'information assiste alors les diverses personnes de l'entreprise - agents opérationnels, managers locaux, concepteurs et stratèges de la DG - en fournissant à chacun la « vue » qui lui convient : ici les données pour le traitement opérationnel d'un dossier ; là les indicateurs utiles au pilotage opérationnel quotidien ; ailleurs les statistiques qui alimentent les études marketing et l'analyse stratégique.

Cette évolution rencontre cependant des obstacles. D'une part, comme l'informatique d'une entreprise résulte d'un empilage historique d'applications conçues dans l'urgence, elle est rarement conforme aux exigences de cohérence du système d'information : il s'en faut de beaucoup que les référentiels et l'administration des données répondent tous aux critères de qualité communément reconnus. D'autre part, l'histoire a habitué les esprits à une représentation étroite de ce que peut et doit être le rôle de l'informatique. Le choc éprouvé lors de l'arrivée des micro-ordinateurs se renouvelle, sous une autre forme, lorsque l'on met en place la documentation électronique, le multimédia et les workflows : personne ne pensait auparavant que l'informatique pouvait ou devait faire cela et il faut du temps pour que l'on *réalise* (aux deux sens du terme) ces nouvelles possibilités. Enfin, l'évolution de l'informatique confronte l'entreprise à des questions qui touchent à son identité même.

* *

La frontière de l'automatisation est l'objet de convictions métaphysiques. Certains pensent qu'en équipant les processus opérationnels on dépasse une limite qui n'aurait pas dû être franchie. Ils éprouvent une horreur instinctive devant le multimédia ou le workflow, horreur qui paraît absurde tant que l'on n'en perçoit pas les raisons profondes. Ainsi s'explique que le même directeur qui lance d'un cœur léger des projets de plusieurs dizaines de millions d'euros refuse un workflow de 100 000 € qu'il considère comme une « usine à gaz ».

Il est vrai qu'il est impossible de tout informatiser : l'informatique doit donc rester en deçà d'une certaine frontière. Mais cette frontière ne passe pas entre le

5. Le langage XML (eXtensible Markup Language), qui permet d'entrelacer du texte et des données structurées, contribue à cette évolution (Elliotte Rusty Harlod et W. Scott Means [34])

Le tracé des frontières dans l'entreprise, question philosophique

D'après le dictionnaire de Lalande ([52] p. 613), l'une des acceptions du mot « métaphysique » est « connaissance de ce que sont les choses en elles-mêmes par opposition aux apparences qu'elles présentent ». On peut utiliser ce terme pour désigner les idées (pertinentes ou non) concernant la nature de l'entreprise ou celle de l'informatique.

Ces idées influencent la façon dont on trace la frontière entre les activités que l'entreprise doit assurer elle-même et celles qu'elle doit sous-traiter. L'intuition des dirigeants étant déconcertée dans les périodes d'innovation, il peut leur arriver d'adopter des principes antiéconomiques.

Certaines entreprises externalisent ainsi leurs centres d'appel (dont la compétence est alors gaspillée) ou encore la maîtrise d'œuvre de leur informatique (ce qui leur fait perdre la maîtrise de leur plate-forme technique), alors qu'elles conservent l'exploitation des serveurs qu'il serait plus efficace de sous-traiter.

concept informatique (légitime) et le processus opérationnel (qu'il ne faudrait pas informatiser) : le traitement de texte, le tableur, la messagerie et la documentation électronique, ainsi que l'Internet et l'Intranet, ont montré que l'informatique pouvait se mettre efficacement non seulement au service des tâches de gestion administratives, mais aussi à celui de l'activité quotidienne de l'agent au travail.

Tout système d'information implique une abstraction, un schématisation, donc le renoncement à la finesse sans limites de l'expérience au bénéfice d'une représentation grossière certes, mais efficace en pratique.

La frontière de l'informatique doit être définie par le degré de détail fonctionnel (et donc conceptuel) qu'il est raisonnable de retenir pour assister l'action des êtres humains, et non par une conception normative (et tissée d'habitudes) du champ légitime de l'informatisation. Le système d'information a été d'abord consacré à quelques fonctions qu'il remplissait en offrant un luxe de fonctionnalités parfois superflues ; il doit aujourd'hui devenir sobre en fonctionnalités, mais s'étendre pour fournir aux utilisateurs l'ensemble des fonctions utilement automatisables.

2.2 Entropie du système d'information

On peut distinguer trois phases dans l'histoire de l'informatique : celle des applications qui ont automatisé les fonctions administratives dans les années 50 et 60 (paie, comptabilité, gestion des stocks etc.) ; celle des systèmes d'information, qui démarre dans les années 70 avec la méthode Merise (mise au point entre 1972 et 1975) ; celle enfin de l'informatisation des processus qui recourt aux langages à objets et généralise dans l'entreprise le « travail assisté par ordinateur » : elle débute dans les années 80⁶.

6. Bernard Morand, professeur à l'Université de Caen, fournit une utile présentation de l'histoire des méthodes (www.iut3.unicaen.fr/~morand/).

Chacune de ces phases peut se représenter par un petit dessin : pour la première, des applications juxtaposées, « en tuyau d'orgue » ; pour la deuxième, qui a ambitionné de corriger le désordre sémantique par la gestion des données de référence, il faut ajouter les bases de données et les référentiels. Un petit diagramme d'activité inspiré d'UML représente convenablement la troisième, où se réalise l'informatisation des processus (figure 2.9).

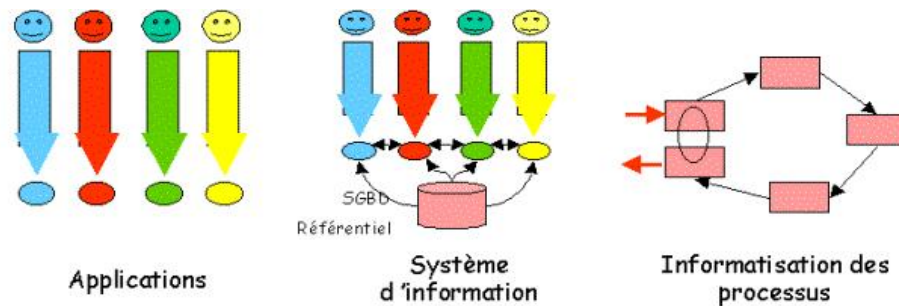


FIG. 2.9 – Les trois phases de l'informatisation

Dans la troisième phase, la conception d'un système d'information doit obéir à quelques principes élémentaires⁷ : bien définir les domaines d'action, les processus de production de valeur ajoutée, les « populations » concernées par ces processus, les « classes » à utiliser pour décrire ces populations ; organiser les processus de façon à éviter les doubles saisies, les doubles identifications, les connexions répétées à des applications diverses ; éliminer les synonymes et les homonymes ; construire les référentiels (identifiants, définitions des données) et gérer les données de référence de sorte que la sémantique du système d'information soit maîtrisée...

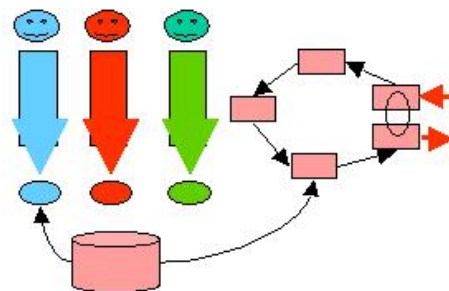
Mais nos entreprises n'ont pas attendu le système d'information ni UML pour s'informatiser ; les applications conçues « en tuyaux d'orgue » dans les années 60 et 70 sont encore là et les refaire coûterait cher⁸. Pour corriger leurs défauts les plus criants on leur a associé des référentiels mais ceux-ci ne recouvrent en général qu'une partie du système d'information (ainsi l'entreprise aura créé un annuaire des personnes et un annuaire de l'organisation, mais non la nomenclature de ses produits etc.) D'ailleurs la construction d'un référentiel pose de difficiles problèmes de méthode (voir page ??).

Les réalisations en mode objets ne concernent aujourd'hui qu'une petite partie des systèmes d'information et doivent se juxtaposer aux réalisations antérieures. Ainsi l'on peut représenter nos systèmes d'information par la figure 2.10 : ils se présentent comme une accumulation géologique de couches diverses obéissant chacune à des priorités et principes différents ; les responsables tentent de se débrouiller pour tirer de cet ensemble disparate la meilleure performance pour le moindre coût.

Lorsqu'on parle de système d'information, il faut indiquer si l'on parle du système d'information tel qu'il existe *de facto* dans une entreprise, souvent conforme à la figure 2.10, ou du système d'information idéal correspondant au cas hypothétique

7. Les principes sont « élémentaires » au plan de la logique, mais cela ne veut pas dire que leur application soit en pratique aisée : en fait elle est rarement réussie ou complète.

8. Le code source de certaines d'entre elles a été perdu : on se garde d'y toucher, en espérant que « cela va continuer à marcher ».

FIG. 2.10 – *Système d'information hybride*

d'une entreprise nouvelle qui partirait de zéro et qui, de plus, ne commettrait aucune erreur de conception.

* *

Même si nous étions parvenus au terme de l'évolution actuelle, même si les systèmes d'information étaient spécifiés en UML et réalisés en mode objets, le désordre y naîtrait aussi naturellement que l'entropie naît dans la matière.

Supposons qu'une entreprise ait créé un référentiel de l'organisation et immatriculé ses services, établissements et zones géographiques. Elle a ainsi construit une base de données de référence qui évolue avec les changements de l'organisation. Tout se passera bien si les divers domaines de l'entreprise répliquent ce référentiel sans délai dans leurs processus ou s'ils le consultent lors de chaque utilisation. Cependant les personnes qui équipent les processus seront toujours tentées de construire un référentiel propre à chaque processus : alors le désordre s'installe.

Les choses se passent souvent ainsi : lors de l'écriture du code, le programmeur introduit dans le programme une copie de la table de référence mais, comme il travaille sous une contrainte de délai, il remet à plus tard l'écriture du module qui assurerait la mise à jour de cette copie. Puis il oublie la nécessité de ce module. Lors de la recette, tout se passera bien puisque la table, étant récente, est à jour. Cependant par la suite la table de référence évoluera. On oubliera parfois de mettre la copie à jour (il faudrait le faire à la main). Alors l'écart se creusera entre les deux tables et le désordre s'installera.

Supposons ainsi que le système d'information comporte *de facto* plusieurs tables représentant le découpage géographique : le monde a été découpé en « régions » et l'entreprise a donné un nom à chacune d'entre elles. Le marché évoluant, l'entreprise modifie ces régions en faisant passer des pays de l'une à l'autre. Le référentiel est modifié ; mais les tables des divers processus ne sont pas mises à jour simultanément. La définition des régions diffère alors d'un processus à l'autre. Les interfaces signaleront des erreurs, les vérifications et redressements accapareront processeurs et back-offices. L'interprétation des données occasionnera, lors des conversations entre dirigeants, d'interminables perplexités.

Si le désordre concerne plusieurs référentiels (des produits, des clients, des pièces détachées etc.), la pagaïe devient générale. Seule une gendarmerie vigilante (ici une *direction de l'architecture* ayant l'information et l'autorité nécessaires au sein de la DSI) peut maintenir la discipline. Cela rappelle la circulation automobile : le code de la route est connu, mais comme chacun peut être tenté de commettre une faute

Isabelle Boydens, *Informatique, normes et temps*

Isabelle Boydens [9] explore les conditions pratiques de production et d'interprétation des bases de données selon une approche à la fois technique et historique. Elle part pour cela d'un cas particulier, la base de données de la sécurité sociale belge. Une telle base de données n'est pas un objet « simple », qu'on la considère en termes de qualité, de représentativité, de pertinence ou de lisibilité.

Isabelle Boydens décrit ainsi sa démarche : « Nous avons préalablement sélectionné un ensemble cohérent et représentatif de normes légales dont la base de données assure l'automatisation. Nous avons ensuite analysé et confronté les sources juridiques, les directives et rapports administratifs, les articles de presse, la documentation informatique et enfin le code de programmation correspondant. Nous avons procédé à de nombreuses entrevues avec les gestionnaires et utilisateurs de la base de données. Enfin nous avons longuement observé le processus de gestion et d'interprétation de la base de données opéré dans la pratique et nous y avons nous même participé. Une observation de terrain permet de révéler ce qui n'est ni dit ni écrit, à savoir les mécanismes informels d'interprétation des données. »

Elle a examiné ainsi des aspects essentiels de la pratique des bases de données, aspects très complexes mais que l'on prend rarement la peine d'étudier parce que l'on suppose, à tort, qu'une base de données est quelque chose de « simple ».

la peur du gendarme est utile.

« Il n'y a qu'à » mettre des gendarmes pour maintenir l'ordre ? Non, car cela ne résout pas tout : parfois les forces de l'ordre sont débordées. Supposez que le système d'information de votre entreprise soit articulé avec celui d'un partenaire (cela suppose l'interopérabilité des deux systèmes d'information) : il faudra s'organiser pour faire prendre en compte par ce partenaire les changements de votre référentiel, et réciproquement. D'ailleurs le système d'information du partenaire sera peut-être en désordre, mais vos gendarmes n'ont pas le « droit de suite » chez lui : s'il utilise pour classer ses produits une table différente par région, vous serez contraint de les connaître toutes et de suivre leurs errements.

Si l'on ne parvient pas à faire prendre au sérieux « ces histoires informatiques » par les dirigeants, s'ils n'en font pas un des thèmes à négocier lors de l'accord de partenariat, le désordre s'insinuera dans le système d'information par les échanges avec les partenaires.

Une autre source de désordre réside dans les changements de périmètre de l'entreprise. Supposez que votre entreprise en achète une autre. L'alignement des systèmes d'information occasionnera des conflits entre équipes de dirigeants (c'est à qui prendra le pouvoir, à qui gardera sa place). Pendant toute la durée de cette guérilla il faut vivre avec un système hétéroclite, des référentiels dont les nomenclatures ne se correspondent pas etc.

Dans l'économie innovante et évolutive d'aujourd'hui les partenariats sont fréquents ainsi que les fusions et absorptions : autant d'occasions pour que l'entropie s'accroisse quelle que soit la qualité des gendarmes. L'état naturel du système d'in-

formation n'est donc pas l'ordre, mais un désordre contre lequel la guerre ne sera jamais gagnée. Ce n'est pas une raison pour perdre de vue les principes selon lesquels on doit bâtir un système d'information, mais il sera en pratique difficile de les respecter exactement.

Comment font les forces de l'ordre lorsqu'elles sont débordées ? Elles louvoient à la recherche du compromis qui permettra le moindre mal : elles pactisent avec une bande pour mettre une autre bande à la raison, elles tolèrent ceci pour pouvoir réprimer cela, elles négocient des appuis auprès de la municipalité, des familles, des associations. Le gendarme se fait diplomate. Il en est de même du directeur de l'architecture quand les sources de désordre ont un fort débit. S'il parvient un instant à imposer la logique, la discipline, la méthode etc., l'ordre sera de courte durée. Il ne pourra pas se contenter de règles formelles : il devra avoir une sensibilité tactique et « politique » pour limiter la casse et faire en sorte que, quoique désordonné, le système d'information reste assez cohérent pour rendre un service acceptable.

2.3 Limites de l'informatique

En informatique, on dit qu'une opération est « complexe » si elle est logiquement possible mais irréalisable en pratique.

Une **première forme** de cette « complexité » provient de la représentation des nombres dans la mémoire de l'ordinateur. Celle-ci ne pouvant contenir qu'une quantité limitée de chiffres, les calculs informatisés portent sur un sous-ensemble des nombres rationnels, approximation des nombres réels. La précision des calculs est donc limitée même si le nombre de chiffres que contient la mémoire est élevé. Il en résulte de grandes difficultés mathématiques⁹.

Une **deuxième forme** de « complexité » est liée au nombre de calculs élémentaires que nécessite une opération. En notant n le cardinal de l'ensemble sur lequel on travaille, on dira que la « complexité » est linéaire si elle demande de l'ordre de n calculs, quadratique si elle en demande de l'ordre de n^2 , « exponentielle » si elle en demande de l'ordre de e^n ou, pire, de n^n . Si par exemple il faut réaliser un calcul élémentaire sur chaque couple d'éléments de l'ensemble, la « complexité » est quadratique car le nombre des couples est égal à $n(n-1)/2$. S'il faut calculer sur chacune des parties de l'ensemble, la « complexité » est exponentielle car le nombre des parties est égal à 2^n . Enfin, si l'on doit faire un calcul sur chacune des permutations des éléments de l'ensemble dont le nombre est $n!$ (factorielle de n), la « complexité » est de l'ordre de n^n car $n! \sim \sqrt{2\pi n}(n/e)^n$ (formule de Stirling).

Certains problèmes à la formulation simple peuvent exiger une durée de calcul

9. « Floating point computation is by nature inexact, and programmers can easily misuse it so that the computer answers consist almost entirely of “noise”. One of the principal problems of numerical analysis is to determine how accurate the results of certain numerical methods will be. There's a credibility gap: We don't know how much of the computer's answers to believe. Novice computer users solve this problem by implicitly trusting in the computer as an infallible authority; they tend to believe that all digits of a printed answer are significant. Disillusioned computer users have just the opposite approach: they are constantly afraid that their answers are almost meaningless. Many serious mathematicians have attempted to analyze a sequence of floating point operations rigorously, but have found the task so formidable that they have tried to be content with plausibility arguments instead » (Knuth [51] vol. 2 p. 229).

de l'ordre de l'âge de l'univers : c'est le cas du « problème du commis voyageur » dès que n atteint quelques dizaines (pour trouver l'itinéraire optimal passant par plusieurs villes il faut calculer la longueur de $n!$ itinéraires)¹⁰.

La « complexité » permet d'évaluer la faisabilité d'un calcul : si n dépasse quelques centaines (c'est le cas de la plupart des bases de données d'une entreprise), un calcul linéaire sera facile, un calcul quadratique difficile et un calcul exponentiel impossible. Le programmeur qualifié arrive parfois à rendre possible un traitement qui, programmé de façon sommaire, aurait été impossible ou difficile : s'il s'agit par exemple de faire un tri, un calcul rustique sera quadratique mais un calcul bien conçu sera d'ordre $n \log(n)$ seulement.

Ces deux formes de « complexité » informatique n'ont rien à voir avec la complexité du réel qu'aucun système fini ne peut décrire. En effet, même si l'opération qui consiste à répéter un grand nombre de fois un calcul élémentaire demande un temps très long, elle n'est pas au plan logique plus complexe que le calcul élémentaire lui-même et celui-ci est aussi simple que l'idée qui a guidé sa conception. Il en est de même pour les difficultés mathématiques liées à la représentation des nombres réels par des nombres rationnels. La « complexité » informatique est un homonyme de la complexité du réel. C'est pourquoi nous avons utilisé les guillemets pour la noter.

Par contre la **troisième forme** de « complexité » provient des limites de la logique elle-même : comme l'a démontré Gödel, aucun système logique ne peut contenir la démonstration de toutes les vérités. Le domaine de la pensée pure est donc lui aussi complexe, puisque aucun système fini ne peut en rendre compte. On peut déduire du théorème de Gödel un des résultats les plus importants de la théorie de l'informatique (Sipser [76] p. 165) : il est impossible de concevoir un programme capable de vérifier tous les programmes.

Supposons en effet qu'un tel programme P existe.

- 1) Si le programme A est juste, $P(A) = v$ (v pour « vrai »).
- 2) Soumettons à P le programme $G = [P(G) = f]$ (f pour « faux »).
- 3) Si $P(G) = v$, le programme $P(G) = f$ est faux ; donc $P[P(G) = f] = P(G) = f$, ce qui est contraire à l'hypothèse.
- 4) Si $P(G) = f$, alors $P[P(G) = f] = P(G) = v$, ce qui est encore contraire à l'hypothèse.
- 5) Ainsi G ne peut pas être vérifié par P .

Il ne peut donc pas exister de programme capable de vérifier tous les programmes. Certes on peut définir des méthodes de vérification efficaces et les outiller de sorte qu'elles soient faciles à utiliser. Mais ces méthodes, aussi efficaces soient-elles, ne garantissent pas que tous les programmes qu'elles valident soient corrects : « Program testing can be used to show the presence of bugs, but never to show their absence » (Edsger W. Dijkstra, « Notes on structured programming », in [24]).

10. Si l'ordinateur fait mille calculs par seconde il faudrait une durée égale à l'âge de l'univers (12 milliards d'années) pour trouver, en calculant tous les itinéraires possibles, le meilleur itinéraire entre 22 villes. Si l'ordinateur fait mille milliards (10^{12}) de calculs par seconde, il faudrait cette même durée pour trouver le meilleur itinéraire entre 28 villes.

Théorème de Gödel

Gödel a démontré que si l'on construit un système logique pour formaliser la théorie des nombres entiers, ce système contient au moins une formule A telle que ni A , ni sa négation $\text{non-}A$ ne peuvent être démontrées dans le cadre de ce système.

Russell et Whitehead [23] avaient tenté de fonder l'ensemble de la logique sur une base axiomatique. Le théorème de Gödel ([37]) prouve que ce but est hors d'atteinte : quel que soit le nombre (fini) des axiomes retenus pour fonder un système logique, il existera toujours des propositions vraies qu'il est impossible de démontrer dans le cadre de ce système.

La démonstration de Gödel est très technique. Voici une description schématique de son raisonnement, tel qu'il le présente lui-même dans l'introduction de son article :

- 1) Supposons qu'il existe une Théorie Complète (TC) fondée sur un nombre fini d'axiomes et permettant, si l'on considère une phrase quelconque, de décider sans jamais se tromper si cette phrase est vraie ou non.
- 2) Considérons la phrase « TC ne dira jamais que la présente phrase est vraie ». Nommons cette phrase G , ce que nous noterons : $G = \text{« TC ne dira jamais que } G \text{ est vraie »}$.
- 3) Soumettons G à TC et demandons à TC de dire si G est vraie ou non.
- 4) Si TC dit que G est vraie, alors G est fausse. Mais alors comme TC a dit que G était vraie, TC a commis une erreur. Cependant par hypothèse TC ne se trompe jamais. Donc TC ne dira jamais que G est vraie.
- 5) Si « TC ne dit jamais que G est vraie », G est vraie. Mais d'après ce que nous venons de voir TC ne pourra jamais le dire.
- 6) Il ne peut donc pas exister de Théorie Complète, c'est-à-dire de théorie permettant, quelle que soit la phrase que l'on considère, de dire si elle est vraie ou non.

Ce raisonnement rappelle le paradoxe fameux mettant en scène un Crétois qui dirait : « Les Crétois ne disent jamais que des mensonges ».

Michael Sipser, *Introduction to the Theory of Computation*

Ce livre présente la théorie de l'informatique : la machine de Turing, la théorie des langages, les algorithmes, la décidabilité, le fait qu'il n'existe pas de programme qui sache vérifier les programmes, le chiffrement à clé publique, les ordres de grandeur qui distinguent le faisable (polynomial) de l'impossible (exponentiel) etc.

Définitions, lemmes et théorèmes s'enchaînent. Sipser est un grand pédagogue. Sans renoncer à la rigueur, il conduit son lecteur le long d'un chemin facile, vérifiable ; en partant d'automates simples il fait découvrir les langages, les grammaires, la machine de Turing, la décidabilité, la récursion, la complexité dans le temps (durée des calculs) et dans l'espace (taille de la mémoire nécessaire) etc. Il est vrai qu'il faut revenir souvent sur ses pas pour se remémorer les résultats acquis, mais c'est le moindre des efforts que l'on puisse consentir.

Sipser écrit deux fois la même démonstration : une fois en langage courant pour donner une idée générale (ce qu'il appelle « proof ideas ») ; la deuxième fois en notation formelle. Lorsqu'on lit un chapitre la première fois, on se contente des « proof ideas » pour avoir une idée d'ensemble. Puis on lit les démonstrations en entier pour se familiariser avec les notations.

2.4 L'ordinateur et l'intelligence

« These machines have no common sense; they do exactly as they are told, no more and no less. This fact is the hardest concept to grasp when one first tries to use a computer. » (Donald E. Knuth, *The Art of Computer Programming* [51] volume 1, p. v)

« (...) es realmente el chimpancé que quiere aprender a leer, un pobre tipo que se da con la cara contra las paredes, y no se convence, y vuelve a empezar. Ah, pero si un día el chimpancé se pone a leer, qué quiebra en masa, qué desparramo, qué sálvese el que pueda, yo el primero. » (Julio Cortázar, *El perseguidor*, Alianza Editorial 1993, p. 63).

Parmi les fausses questions que conduit à poser un concept mal bâti se trouve celle de l'« intelligence » des ordinateurs. Elle suscite des émotions extrêmes et opposées. « Où est l'homme là-dedans ? » demande avec angoisse l'individualiste sentimental qui croit être un humaniste (Virilio [84]). Certains, tout aussi émotifs mais misanthropes, se félicitent de voir la machine éliminer l'être humain qu'ils jugent peu fiable et moralement douteux (Truong [82]). D'autres enfin croient devoir utiliser un vocabulaire emphatique pour célébrer la nouveauté qu'apporte l'ordinateur (Lévy [55]). D'honorables philosophes¹¹ et d'excellents ingénieurs¹² disent ainsi des choses qui

11. « Aujourd'hui notre mémoire est dans le disque dur. De même, grâce au logiciel, nous n'avons plus besoin de savoir calculer ou imaginer. L'humain a la faculté de déposer les fonctions de son corps dans les objets. Et il en profite pour faire autre chose. » (Michel Serres dans *L'Expansion*, 20 juillet 2000).

12. « Dans 10 à 20 ans, l'ensemble des ordinateurs interconnectés aura des capacités intellectuelles qui dépasseront l'homme dans tous les domaines. » (Jean-Paul Figer)

nous semblent ne pas rendre un compte exact du phénomène de l'informatisation.

Pour éclairer la question il est utile de se référer de nouveau à la culture américaine. Nous considérerons ici non plus l'entreprise, mais l'armée (Desportes [20] p. 155) : tout comme l'entreprise, l'armée est consacrée à l'action mais sous sa forme la plus dangereuse. C'est donc dans les doctrines militaires que l'on rencontrera l'expression la plus claire de la doctrine d'action d'une Nation, doctrine qui s'appliquera ensuite dans tous les domaines et en premier dans l'entreprise, lieu géométrique de l'action organisée.

La doctrine militaire américaine s'appuie sur le *Précis de l'art de la guerre* [48] du Suisse Antoine Jomini (1779-1869) et, à un moindre degré, sur le *Vom Kriege* [87] du Prussien Carl von Clausewitz (1780-1831). Ces deux généraux avaient observé la stratégie de Napoléon, étudié celle de Frédéric II de Prusse, et en avaient tiré les leçons. Rivaux sur le terrain de la théorie, ils se sont beaucoup emprunté l'un à l'autre. Leurs théories sont donc plus proches que les interprétations qui en sont données : mais, quand il s'agit d'évaluer une influence, l'interprétation d'une théorie a plus de poids que son contenu même. On peut ainsi associer les noms de Jomini et Clausewitz à deux modèles contrastés.

Selon le modèle de Jomini, tout problème doit être ramené à ses éléments rationnels puis traité de façon scientifique : Jomini entend maîtriser l'art de la guerre en résumant ce qu'elle a de complexe par quelques principes et axiomes évidents. Cette approche satisfait le pragmatisme des Américains : culte de l'offensive, concentration des masses au point décisif pour anéantir l'adversaire, professionnalisation d'une stratégie qui a pour but de gagner les batailles et non de servir par les armes un projet politique. La guerre est un *job* que la nation confie au militaire et que celui-ci exécute.

L'école qui se réclame de Clausewitz ne répudie pas cette modélisation mais postule une continuité entre la guerre et la politique, idée difficilement assimilable par le pragmatisme américain. Cette école insiste par ailleurs sur les conditions de l'action, ces « incertitudes », ces « frictions », ce « brouillard » que Jomini suppose négligeables mais qui en pratique nécessitent pour éclairer la décision un travail permanent d'interprétation et de synthèse.

Le modèle de Jomini, rationnel et planificateur, incite à l'automatisation parfaite : les êtres humains exécutent les ordres de l'automate pré-programmé qui, sur la base d'une information claire et complète, établit de façon optimale le programme de leur action. À la limite, le stratège n'est rien d'autre que le porte-voix de l'ordinateur. Mais le modèle de Clausewitz, plus complexe, invite au contraire à articuler l'automate et l'être humain car seul ce dernier, s'il a été bien formé, pourra dans une situation imprévue interpréter des rapports incomplets ou fallacieux et prendre la décision juste.

* *

La discussion sur l'intelligence de l'ordinateur s'éclaire si on la situe sur cette toile de fond doctrinale. Soit on suppose les hypothèses du modèle de Jomini vérifiées (information parfaite, planification parfaite etc.) et alors l'action peut être préparée par un programme et déterminée au mieux par l'automate, car il calcule très vite et sans erreurs. Soit on suppose au contraire que la situation comporte une part d'incertitude, de « brouillard », et alors la contribution du cerveau humain est

indispensable.

De ces deux modèles, lequel est le bon ? Lorsqu'il s'agit d'assurer la maintenance d'un équipement, le modèle de Jomini semble s'imposer (sous la seule réserve que le technicien émette une alarme en cas d'imprévu). Mais si l'ordinateur permettait de supprimer l'incertitude, cela se saurait dans les salles de marché : les informaticiens auraient fait fortune à la Bourse, et du coup la Bourse aurait cessé d'exister ¹³.

Si nous reprenons la distinction proposée au début de ce chapitre entre « événement interne » et « événement externe », on peut dire que le modèle de Jomini s'applique aux événements internes, à la conduite du processus de production de l'entreprise, et que le modèle de Clausewitz s'applique aux événements externes, que l'entreprise ne peut ni maîtriser ni organiser entièrement : relations avec les clients et avec les fournisseurs, conditions de la concurrence.

Sur ces événements externes, et même si l'information sur le passé et le présent était parfaite - ce qui n'est jamais le cas - l'anticipation comporterait encore une part d'incertitude. Les conditions pratiques de l'action stratégique, celle qui porte non sur l'exécution des tâches mais sur leur conception, sur le positionnement de l'entreprise, sont donc celles que décrit le modèle de Clausewitz. Le modèle de Jomini conforte la confiance en soi du professionnel mais si celui-ci l'applique aux événements externes il prend le risque de se trouver désarmé devant l'imprévu.

* *

Ceux qui conçoivent les ordinateurs savent faire la différence entre l'ordinateur et l'être humain. John von Neumann ¹⁴ était arrivé à l'informatique par la physique et se servait de l'ordinateur pour résoudre par simulation des problèmes trop complexes pour être traités analytiquement. C'est lui qui, avec Church et Turing, a conçu les ordinateurs dont nous nous servons aujourd'hui. Certaines de ses réflexions sont à l'origine de l'intelligence artificielle.

Le cerveau lui a fourni une image de l'ordinateur : tout comme le cerveau, l'ordinateur est doté d'une mémoire, d'une capacité de traitement et d'organes sensoriels (entrées et sorties). Le cerveau, être biologique, progresse à l'échelle des dizaines de milliers d'années, alors que l'ordinateur, être artificiel, progresse à l'échelle de la dizaine d'années, donc beaucoup plus vite. Certains en ont conclu que l'ordinateur, « cerveau électronique », avait vocation à supplanter le cerveau humain. « Comment pouvez-vous affirmer, disent-ils, que l'ordinateur sera à jamais incapable d'avoir des émotions et de se comporter comme un être humain ? »

Du point de vue poétique toute parole qui éveille l'intuition est légitime, car la poésie relève d'une démarche mentale antérieure au raisonnement. Mais certaines phrases sont impropres à alimenter le raisonnement parce qu'elles ne sont pas « falsifiables », pour utiliser le vocabulaire de Popper. « Y a-t-il une vie après la mort ? » La réponse peut être « oui » ou « non » sans que l'expérience puisse trancher, et la conviction intime du croyant lui-même oscille entre ces deux réponses. Donc même si cette question est un thème intéressant pour la rêverie, du point de vue de la

13. Tout comme les Casinos auraient fermé si en physique le déterminisme ne comportait pas d'exception.

14. von Neumann, [89]. *The Computer and the Brain* est le dernier ouvrage de von Neumann. Miné par le cancer dont il est mort en 1957, il n'a pas pu l'achever. Il lui accordait beaucoup d'importance.

réflexion elle est futile. « L'intelligence des ordinateurs atteindra-t-elle, dépassera-t-elle celle des êtres humains? » est une phrase du même type. On peut y répondre par oui, par non, ou osciller entre les deux réponses sans pouvoir trancher par l'expérience puisque celle-ci se situe dans un futur indéfini et ne peut s'appuyer sur aucun précédent.

Il existe entre l'ordinateur et l'être humain une différence bien plus grande qu'entre l'être humain et les primates qui lui sont génétiquement proches. Se peut-il qu'un chimpanzé apprenne un jour à parler ou à lire (voir la citation de Cortázar page ??)? Cette hypothèse n'est ni plus ni moins plausible que celle de l'intelligence de l'ordinateur.

Hypothèse pour hypothèse, nous sommes libres de choisir. Ne serait-il pas alors plus fécond de postuler qu'il existe entre l'ordinateur et le cerveau humain une différence de nature, une différence infranchissable? Cette dernière hypothèse à plusieurs avantages : elle est conforme à l'expérience présente car aucun des ordinateurs existants ne se comporte comme un être humain ; elle indique une piste à l'action : si l'ordinateur et l'être humain sont différents, il importe de les articuler ; enfin, elle fournit à la pensée un cadre utile pour préparer cette action : il est possible de penser l'articulation de deux êtres différents, alors qu'il serait impossible de penser l'articulation de deux êtres que l'on suppose identiques, parce que la pensée les confond.

Von Neumann a perçu les limites de l'analogie entre l'ordinateur et le cerveau humain. Le mécanisme du cerveau, dit-il, est plus lent et davantage sujet à l'erreur que celui de l'ordinateur ; mais la structure du cerveau est, elle, plus riche que celle de l'ordinateur. Le cerveau met en œuvre des processus parallèles et analyse les signaux de façon statistique, notamment en évaluant des corrélations. Cela lui confère une grande rapidité et une grande fiabilité malgré les défauts que comporte son mécanisme. En outre la mémoire du cerveau, qui utilise une grande diversité de supports, semble pratiquement infinie même si la mémoire consciente est, à chaque instant, limitée.

Ainsi notre héritage génétique nous aurait doté d'un automate naturel dont les capacités diffèrent essentiellement de celles d'un ordinateur. Notre cerveau est supérieur à l'ordinateur pour certaines fonctions (interpréter, synthétiser, comprendre, expliquer, décider, concevoir), inférieur et même très inférieur pour d'autres (classer et trouver une information, calculer, recopier, transcoder). La question pratique à laquelle nous sommes confrontés n'est pas de mettre au point un ordinateur dont le comportement ne différerait en rien de celui d'un être humain - but peut-être hors de portée et en tout cas éloigné des priorités actuelles - mais de réaliser efficacement l'articulation entre l'« être humain organisé » (car bien sûr il ne suffit pas de considérer un seul cerveau : il faut considérer la société entière avec ses institutions et, en particulier, l'entreprise qui est le lieu même de l'organisation) et l'« automate programmable doué d'ubiquité » (forme que l'ordinateur a prise grâce au réseau).

2.4.1 Turing a-t-il perdu son pari?

Dans un article qui a eu une immense influence (Turing [83]) Alan Turing a soutenu qu'il était possible de concevoir une expérience prouvant que l'intelligence de l'ordinateur ne pouvait pas être distinguée de celle d'un être humain. Le *pari de*

Turing a éveillé l'ambition de l'intelligence artificielle.

Pour répondre à la question « est-ce que les machines peuvent penser ? » il faut bien sûr pouvoir faire abstraction de l'apparence physique. *Turing* propose donc un « jeu de l'imitation » qu'il définit ainsi :

« Le jeu de l'imitation se joue à trois personnes, un homme (A), une femme (B) et un interrogateur (C) qui peut être de l'un ou l'autre des deux sexes. L'interrogateur se trouve dans une pièce séparée des deux autres. Le but du jeu est pour l'interrogateur de deviner lequel de ses deux interlocuteurs est un homme et lequel est une femme. Il les nomme X et Y, et à l'issue du jeu il dit soit "X est A et Y est B", soit "X est B et Y est A". Il peut poser des questions à A et B (...) Maintenant nous nous demandons "Que peut-il arriver si l'on fait tenir par une machine le rôle de A dans ce jeu ?" Est-ce que l'interrogateur se trompera aussi souvent que lorsque la partie se joue entre un homme et une femme ? Ces questions remplacent notre question initiale, "Des machines peuvent-elles penser ?" (...) Je crois que dans cinquante ans environ il sera possible de programmer des ordinateurs ayant une mémoire de l'ordre de 10^9 de telle sorte qu'ils jouent tellement bien au jeu de l'imitation qu'un interrogateur moyen n'aura pas plus de 70 chances sur cent de les identifier de façon exacte après les avoir questionnés pendant cinq minutes. (...) La seule façon satisfaisante de prouver cela, c'est d'attendre la fin du siècle et de faire l'expérience que je viens de décrire. »¹⁵

Pour comprendre la nature du test de *Turing*, il faut réfléchir un instant à son énoncé. Si la différence entre A et B est évidente l'interrogateur ne se trompera jamais : la probabilité qu'il ne fasse pas d'erreur est donc égale à 1. Si la différence entre A et B est insensible, l'interrogateur se trompera une fois sur deux (il faut supposer qu'en cas de doute il tire sa réponse à pile ou face) : la probabilité qu'il ne fasse pas d'erreur est alors égale à 0.5.

Le test peut donc être caractérisé par la fréquence des résultats exacts, qui appartient à l'intervalle $[0.5, 1]$ (figure 2.11). *Turing* ne dit pas que le test sera réussi si cette fréquence est de 0.5, valeur qui correspond au cas où l'on ne pourrait pas distinguer l'ordinateur de l'être humain : il dit que le test sera réussi si cette fréquence est comprise dans l'intervalle $[0.5, 0.7]$, c'est-à-dire *si l'interrogateur a confondu l'ordinateur avec un être humain dans au moins 30 % des cas*.

Ce test est peu exigeant : il ne dure pas plus de cinq minutes et le seuil d'efficacité est modeste. Il est donc audacieux de prétendre qu'une telle expérience, si elle

15. « [The imitation game] is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A". The interrogator is allowed to put questions to A and B (...) We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?" (...) I believe that in about fifty years' time it will be possible to programme computers, with a storage capacity of about 10^9 , to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning. (...) The only really satisfactory support that can be given for [this] view (...) will be that provided by waiting for the end of the century and then doing the experiment described. »

FIG. 2.11 – *Échelle du test de Turing*

réussissait, autoriserait à affirmer que des machines puissent penser.

* *

Quoiqu'il en soit, Turing a formulé à la fois un pari et le test qui permet de le vérifier. Nous pouvons faire le test, puisque la fin du siècle est passée. Certes l'ordinateur s'est révélé redoutable au jeu d'échecs ; mais dans un travail aussi « simple » que la traduction d'un texte littéraire il fournit un résultat tellement médiocre que l'examineur moyen le distinguera immédiatement d'un traducteur humain.

Si nous prenons Turing au mot, nous pouvons donc dire qu'il a perdu son pari. Mais ce serait là une réfutation peu satisfaisante : quelqu'un d'autre pourra prendre la relève et parier de nouveau en fixant comme échéance la fin du XXI^e siècle ou celle du troisième millénaire, ce qui laisserait la question en suspens pendant un long délai.

La meilleure réfutation de Turing s'appuie en fait sur la définition de l'ordinateur qu'il donne lui-même :

« On peut expliquer l'idée qui se trouve derrière les ordinateurs numériques en disant qu'ils sont conçus pour réaliser toutes les opérations qui pourraient être faites par un calculateur humain. Le calculateur humain est supposé suivre des règles fixes ; il n'a pas le droit de s'en écarter le moins du monde. Nous pouvons supposer que ces règles lui sont fournies dans un livre qui est modifié chaque fois qu'on veut lui faire faire un nouveau travail. Il dispose aussi pour faire ses calculs d'une quantité illimitée de papier. Il peut aussi faire ses additions et multiplications sur une machine à calculer, mais c'est sans importance. Le livre de règles qu'utilise notre calculateur humain est évidemment une fiction commode : en fait, les vrais calculateurs humains se rappellent ce qu'ils ont à faire. Si on veut faire imiter par une machine le comportement d'un calculateur humain, on n'a qu'à demander à celui-ci comment il fait, puis traduire sa réponse en une table d'instructions. Construire une table d'instruction, c'est ce que l'on appelle "programmer". "Programmer une machine pour réaliser l'opération A" équivaut à introduire dans la machine la table d'instructions qui convient pour qu'elle réalise A. »¹⁶

16. « The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer. The human computer is supposed to be following fixed rules; he has no authority to deviate from them in any detail. We may suppose that these rules are supplied in a book, which is altered whenever he is put on to a new job. He has also an unlimited supply of paper on which he does his calculations. He may also do his multiplications and additions on a "desk machine," but this is not important. The

Si Turing définit ainsi l'ordinateur, puis affirme par ailleurs qu'il sera difficile de distinguer au « jeu de l'imitation » un ordinateur d'un être humain, cela implique qu'il sera tout aussi difficile de distinguer un calculateur humain, assujetti au respect de règles fixes consignées dans un cahier, d'un être humain ordinaire qui, n'étant pas assujetti à de telles règles, sera libre de prendre des initiatives et de s'adapter à l'imprévu.

Il s'agit donc finalement de savoir si l'on suppose que l'imprévu est négligeable ou non : on retrouve ainsi le débat entre les deux écoles de pensée qui s'inspirent l'une de Jomini, l'autre de Clausewitz. Il est possible de rédiger le cahier de règles de sorte que le calculateur dispose de consignes lui permettant de faire face à une grande diversité de situations (c'est ainsi que l'ordinateur joue aux échecs), mais la typologie des situations que décrit le programme est nécessairement limitée alors que l'imprévu potentiel que comporte le rapport d'un être humain avec le monde de la nature est *indéfini* : l'être humain doit savoir « se débrouiller » lorsqu'il rencontre un imprévu, là même où l'ordinateur (ou le calculateur humain, pur exécutant) ne disposerait pas de règles et ne saurait donc plus quoi faire, ou commettrait des erreurs : c'est ce type de difficulté que l'ordinateur rencontre en traduction automatique.

C'est le sens de la réfutation de Turing par Popper¹⁷ (Popper [67]) :

« Turing a affirmé qu'il était impossible, par principe, de distinguer les hommes des ordinateurs à partir de leurs activités observables (leurs comportements), et il a défié ses adversaires de spécifier une activité ou un comportement humain observable quelconque qu'un ordinateur fût, par principe, incapable de réaliser. Mais ce défi est un piège intellectuel : spécifier un type de comportement reviendrait à établir une spécification pour la construction d'un ordinateur. En outre, si nous utilisons et construisons des ordinateurs, c'est parce qu'ils sont capables de faire beaucoup de choses que nous ne savons pas faire ; tout comme je me sers d'un crayon ou d'un stylo pour calculer une addition que je suis incapable de faire de tête. “Mon stylo est plus intelligent que moi”, disait souvent Einstein. Mais ceci ne prouve pas qu'il soit impossible de le distinguer de son stylo. »

* *

Turing fut un génie et aussi un être humain dont les souffrances suscitent la compassion¹⁸. Il a contribué à la naissance de l'informatique, bouleversement dont nous n'avons pas fini d'explorer les conséquences. Il est compréhensible que le témoin d'un événement d'une telle importance ait eu du mal à en circonscrire la portée. Constatant la rapide croissance des possibilités de l'ordinateur Turing a extrapolé,

book of rules which we have described our human computer as using is of course a convenient fiction. Actual human computers really remember what they have got to do. If one wants to make a machine mimic the behavior of the human computer in some complex operation one has to ask him how it is done, and then translate the answer into the form of an instruction table. Constructing instruction tables is usually described as “programming”. To “programme a machine to carry out the operation A” means to put the appropriate instruction table into the machine so that it will do A. »

17. Popper, comme d'autres commentateurs, pousse à l'extrême la formulation du test de Turing qui est comme nous l'avons vu modeste. Mais ce n'est que justice puisque Turing a prétendu que la réussite du test permettrait d'affirmer que les machines « pensent ».

18. Homosexuel, Alan Turing (1912-1954) fut comme Oscar Wilde (1854-1900) une victime du système judiciaire britannique. Il s'est très vraisemblablement suicidé (Hodges [44]).

et il s'est trompé.

Les Grecs avaient découvert la puissance de l'abstraction ; on voit dans *L'Anabase* de Xénophon l'ivresse intellectuelle que cette découverte a provoquée. Platon a été jusqu'à affirmer que seules les idées étaient réelles : l'idée de cheval serait ainsi plus réelle que le cheval qui gambade dans le pré tout comme l'idée de cercle est plus réelle que le cercle que l'on trace avec un compas. L'idéalisme platonicien a inspiré la philosophie de Plotin et de saint Augustin, la théologie catholique, les hommes de la Renaissance et bien d'autres : ce n'est pas parce qu'une représentation est inexacte ou incomplète qu'elle n'a pas d'impact, surtout quand elle est formulée par une personne de grand talent.

Le pari de Turing a eu des effets contrastés. Le parallèle entre le cerveau électronique et le cerveau humain a suscité des recherches fécondes, même si les travaux en intelligence artificielle ont donné des résultats décevants. Mais en focalisant l'attention sur l'intelligence de l'ordinateur et donc sur la *similitude* entre celui-ci et l'être humain, ce pari a eu l'inconvénient d'inciter à ne pas se soucier de leurs *différences*.

Quelles sont nos priorités *aujourd'hui* ? Est-ce de rêver à l'intelligence que la machine, par hypothèse, pourrait déployer dans quelques siècles ou quelques millénaires, ou d'agir pour articuler au mieux les ressources de l'automate programmable (l'ordinateur) avec ce que sait faire l'être humain organisé dans l'entreprise (comprendre, expliquer, décider, concevoir) ? Choisir la seconde réponse, c'est dire que le pari de Turing n'était pas pertinent.

* *

Les questions que nous venons d'évoquer n'ont rien de fondamentalement nouveau. Le cerveau humain s'est déjà accommodé de plusieurs « successeurs » potentiels : le langage, qui a une existence propre depuis que des hommes se sont mis à parler ; l'écriture, support de la mémoire ; l'impression, support de la diffusion des textes. Des machines remplacent nos jambes (bateau, bicyclette, automobile, avion), des prothèses assistent nos sens (lunettes, appareils acoustiques, téléphones, télévision). L'élevage et l'agriculture pratiquent depuis le néolithique la manipulation génétique par sélection des espèces. La bionique, l'intelligence artificielle ne font que venir s'ajouter au catalogue des prothèses qui assistent nos activités physiques ou mentales. N'avons nous pas tendance, par défaut de perspective historique, à exagérer leur nouveauté ?

Il existe entre la complexité du logiciel et celle du cerveau d'un être vivant une différence de nature. Aussi compliqué soit-il, le logiciel est de taille finie puisqu'il s'agit d'un *texte*. Mais toute théorie, aussi puissante soit-elle, reste incomplète ; aucun objet naturel (et le cerveau en est un) ne peut donc être reproduit par un texte. Si un texte poétique semble parfois nous mettre en relation avec le monde lui-même, c'est parce que notre cerveau le complète par le réseau de connotations qui enrichit l'apport des mots et, au prix d'une imprécision que le logiciel ne saurait tolérer, ouvre à notre rêverie une perspective indéfinie.

L'intuition de ceux qui vivent dans un univers de science fiction ou de dessin animé s'affranchit de l'expérience : dans cet univers-là toutes les métamorphoses sont possibles, toutes les chimères peuvent exister, quiconque évoque une impossibilité serait immédiatement démenti. Mais l'expérience distingue le possible de l'impossible et assigne des bornes à notre action. Il existe dans le monde de l'expérience

des questions *pertinentes*, c'est-à-dire utiles à l'action, d'autres qui ne le sont pas. Si elles nous détournent des questions pertinentes, nous devons qualifier les rêveries sur l'intelligence des ordinateurs d'*impertinentes*.

* *

Nous qui savons tant bien que mal parler, lire, écrire, compter, domestiquer plantes et animaux, fabriquer produits et outils, communiquer, déposer et retrouver notre mémoire collective dans des encyclopédies etc., devons apprendre à tirer parti de l'automate programmable. Pour voir clair dans les questions de savoir-faire et de savoir-vivre, d'organisation collective et personnelle que cela pose, il importe de percevoir la frontière qui nous sépare de lui, de discerner ce qu'il sait faire de ce que nous savons faire, de sorte que son insertion dans notre action, dans nos processus, puisse être judicieuse. Il faut pour tracer cette frontière un outil conceptuel aussi précis que le scalpel du chirurgien.

Or les rêveries sur l'intelligence de l'ordinateur brouillent cette frontière. On ne peut pas penser la relation entre deux êtres dont on a postulé l'identité, fût-elle asymptotique. L'intelligence de la machine s'actualisant dans un futur indéfini, l'intuition s'évade des contraintes de l'action et tourne le dos aux questions qui aujourd'hui sautent aux yeux : comment assister nos processus opérationnels ; tirer parti de la conjugaison des données et du commentaire ; fonder la solidité des référentiels ; articuler les médias ; faire interopérer les systèmes d'information de diverses entreprises ; assurer la dialectique du système d'information et de la stratégie etc.

Ceci n'est pas sans conséquences. La rêverie est innocente quand elle délasse le praticien expert : il n'est pas dupe des illusions qu'elle comporte et il est lié à l'expérience par le ressort de rappel de son activité professionnelle. Mais parmi les personnes qui décident en matière de système d'information les praticiens experts sont une minorité. La diffusion médiatique des rêveries sur l'intelligence de l'ordinateur risque de placer les personnes sur une orbite mentale d'où il leur sera impossible de revenir au sol - et alors leurs décisions seront follement inadéquates. Ce n'est donc pas de rêveries impertinentes que nous avons besoin dans ce domaine difficile, mais de démarche scientifique et de méthode expérimentale.

L'ordinateur ne sait pas réaliser des choses que l'être humain apprend à faire lors des premières années de sa vie : il ne comprend pas le langage humain ordinaire, avec ses allusions et ses connotations. Il ne sait pas faire la synthèse d'un ensemble de faits et en tirer la conclusion. Il ne sait pas prendre de décision. Il n'a pas d'imagination. Si l'on a l'impression qu'il sait faire tout cela, c'est que l'on commet une erreur classique : celle qui consiste à dire « l'ordinateur calcule » quand on se sert de l'ordinateur pour faire un calcul, ou que « l'ordinateur décide » quand on l'utilise pour aider la décision.

Ce n'est pas l'ordinateur qui calcule ni qui décide, mais l'utilisateur qui se fait aider par l'ordinateur pour calculer ou décider. Cette aide rend parfois possibles des choses qui, autrement, auraient été impossibles : ainsi le pilote automatique d'un avion permet de maintenir celui-ci pendant toute la durée du vol dans la position qui minimise la consommation de carburant, alors pour le pilote humain ce serait aussi délicat que de tenir une assiette en équilibre sur une aiguille, performance qu'il est difficile de réaliser des heures durant. Les écrivains de science fiction, les cinéastes, créent un monde imaginaire ; il leur est facile d'y doter les ordinateurs de facul-

tés extraordinaires, comme l'a fait Stanley Kubrick dans *2001 : odyssée de l'espace*, ou de mettre en scène des robots qui se comportent comme des êtres humains. Ils sculptent ainsi un imaginaire fallacieux¹⁹.

* *

Pour progresser dans l'utilisation de l'ordinateur, il faut se rappeler qu'il obéit à la lettre, sans initiative ni interprétation, aux ordres qui lui sont donnés ; il apporte à l'être humain une aide précieuse, mais bien délimitée. L'expression « assisté par ordinateur », que l'on utilise pour le dessin, la gestion, la conception etc., a une portée très générale : en tout et pour tout, l'ordinateur nous assiste ; les utilisations les plus intelligentes de l'informatique sont celles qui cultivent cette relation entre l'ordinateur et nous, en partant d'une claire conscience de la différence qui nous sépare de lui.

2.4.2 Articuler l'ordinateur et l'être humain

Pour concrétiser le propos, considérons quelques exemples :

Une entreprise veut mettre sur son site Web la fonction d'intermédiation assurée jusqu'ici par une « première ligne » proche des clients et très expérimentée. Le client déposera sa demande, les fournisseurs leurs offres, l'informatique fera le rapprochement entre les deux. Mais il faut mettre l'entreprise en garde. Que va-t-il se passer si elle confie l'intermédiation à un automate ? Ne va-t-elle pas y perdre en efficacité ?

On sait qu'un logiciel de traduction automatique est (1) coûteux à concevoir, (2) fécond en contresens insupportables à la lecture, (3) moins efficace qu'un logiciel de traduction assistée par ordinateur dont la production coûte mille fois moins cher. Si vous souhaitez que l'ordinateur fasse tout, vous devez réaliser un programme qui saura traiter les cas particuliers ; il sera d'une complexité monstrueuse et en fait il ne parviendra pas à traiter *tous* les cas particuliers. Si vous articulez l'ordinateur et l'être humain, vous pouvez laisser à ce dernier le soin d'interpréter les cas particuliers en utilisant son discernement et son jugement. L'ordinateur traitera les cas simples, qui constituent la majorité des affaires ; son programme sera plus simple, donc moins coûteux, plus évolutif, mieux maîtrisable.

Il se produit des catastrophes lorsque le système d'information est conçu de telle sorte que les personnes de la première ligne n'aient pas la possibilité d'agir alors qu'elles ont expérience et bon sens et comprennent correctement chaque cas particulier. Il est impossible d'imaginer à l'avance la gamme des incidents qui peuvent survenir en cas d'automatisation, mais il est certain que des incidents se produiront et l'absurdité peut se déployer sans limites si on ne ménage pas la possibilité d'une « reprise de main » par un être humain.

* *

19. « Contrairement à ce que l'on voit dans les films, les logiciels qui existent dans le monde réel sont incroyablement primitifs si on les confronte aux exigences de ce que nous appelons le "simple bon sens". Regardez par exemple la stupidité incroyable des moteurs de recherche sur le Web lorsqu'on leur demande de trouver des contenus ayant un sens pour l'être humain » (Bruce Schneier, « Semantic Network Attacks », *Communications of the ACM*, décembre 2000).

La traduction automatique

La traduction automatique ne donne de résultat acceptable que lorsqu'il s'agit de textes formalisés, techniques, précis, sans surprises du point de vue de la syntaxe. Par contre les textes littéraires, proches de la langue ordinaire et riches en connotations, donnent des résultats ridicules lorsqu'ils sont soumis à la traduction automatique.

Pour voir ce qu'il en est, appliquons le logiciel fourni par AltaVista à un paragraphe d'une fameuse lettre ouverte de Bill Gates(Gates[36]) :

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

On obtient ceci :

À moi, la chose la plus critique sur le marché de passe-temps est en ce moment le manque de bons cours, de livres et de logiciel de logiciel lui-même. Sans bon logiciel et propriétaire qui comprend programmant, un ordinateur de passe-temps est gaspillé. Le logiciel de qualité sera-t-il écrit pour le marché de passe-temps?

Certes, il s'agit là d'un cas particulier et il existe des logiciels de traduction plus puissants (notamment ceux que l'on étalonne par analyse statistique d'un corpus de textes). Toutefois le risque d'un contresens non seulement grossier, mais absurde, sera toujours élevé.

Il est par contre possible d'écrire un bon logiciel de *traduction assistée par ordinateur* qui aidera efficacement un traducteur connaissant bien la langue de départ et très bien la langue d'arrivée. Mais les producteurs de logiciels et les organismes de recherche ont préféré dépenser des fortunes pour la traduction automatique au lieu de chercher à aider les traducteurs.

Il existe une façon raisonnable d'utiliser la traduction automatique. L'automate peut balayer un ensemble de textes écrits dans une langue que l'on ignore, en fournir une transcription sommaire et aider ainsi à repérer ceux qui méritent d'être effectivement traduits ; puis la traduction proprement dite sera faite par traducteur humain assisté par l'ordinateur. On est loin, avec cette utilisation modeste, des ambitions initiales de la traduction automatique.

Il est vrai que l'articulation entre l'automate et l'être humain demande des consultations, de la réflexion, bref une *démarche* délicate.

Démarche de mise en place de l'e-business

« Il ne suffit pas pour votre entreprise d'avoir une présence sur le Web, fût-elle jolie. Il faut d'abord connaître vos clients et savoir ce qu'ils attendent de vous car le Web, c'est le pouvoir au client : si vous ne répondez pas comme il le souhaite, clic ! il est parti, vous pouvez lui dire adieu.

« Quel positionnement voulez-vous donner à votre entreprise sur le Web ? jusqu'où voulez-vous pousser la différenciation de votre offre ? avec quels partenaires voulez-vous vous associer ? quelles relations souhaitez-vous avoir avec vos fournisseurs ? jusqu'où entendez-vous pousser l'intégration entre vos affaires et celles de vos partenaires, fournisseurs et clients ? Il faut ici une ingénierie d'affaire, avec ses dimensions juridique et financière.

« Souhaitez-vous conserver la même périphérie, ou pensez-vous qu'il faut externaliser certaines de vos activités ? Le e-business va de pair avec un e-management : il faut repenser la personnalité, les priorités, les contours de l'entreprise. Il convient que cette réflexion ne soit pas seulement celle du PDG mais qu'elle soit partagée par les managers, les cadres, finalement par toute l'entreprise : cela suppose des consultations, concertations et validations.

« Enfin quand vous saurez ce que vous voulez faire il faudra s'assurer que c'est faisable. Vos limites sont ici celles de votre système d'information. Si celui-ci est constitué d'une accumulation d'applications hétéroclites fondées sur des définitions incohérentes, si les données de référence ne sont pas gérées, s'il n'existe pas de gestion de configuration, bref si vous n'avez pas un système d'information digne de ce nom, vous aurez du mal à jouer la partie de l'e-business. Ce ne sera pas totalement impossible - il n'est pas indispensable de passer par un ERP^a avant de se lancer sur l'Internet - mais ce sera difficile. Le calendrier de mise en œuvre des fonctionnalités de l'e-business sera articulé avec la mise à niveau de votre système d'information.

« Vous pouvez démarrer tout de suite, mais il vous faudra quelques années pour transformer l'entreprise. »

^a « Enterprise Resource Planning »

Certaines personnes disent alors : « Vous proposez une démarche, mais ce qu'il nous faut c'est un *produit* ». Les fournisseurs de logiciels portent une part de responsabilité dans cette erreur de perspective. On a vu, sur la couverture du *Monde Informatique* (n° 839 du 4 février 2000) la photo d'un fromager accompagnée de cette légende : « De quoi avez-vous besoin pour transformer votre business en e-business ? Découvrez-le page 11 ». Et à la page 11 se trouvait une publicité contenant ces mots : « Il faut un puissant logiciel pour transformer le business en e-business. Ce logiciel existe, IBM l'a fait » Or la première question qu'une entreprise doit se poser n'est pas « quel logiciel choisir », mais « que veulent mes clients », puis « quel rôle dois-je jouer » etc. La check-list ne commence pas par la technique mais par la stratégie. En suggérant qu'il suffit de prendre un logiciel - le sien bien sûr - IBM oriente ses

clients vers la voie de l'échec même si son logiciel est excellent (et IBM a d'excellents produits).

Les entreprises n'ont que trop tendance à croire que tout problème est technique (c'est-à-dire relève étymologiquement du savoir-faire) et donc que toute solution doit être également technique. Mais avant de savoir faire, il faut savoir ce que l'on veut faire, pourquoi et pour qui on veut le faire. « Pourquoi faire » et « vouloir faire » doivent précéder « savoir faire », sans quoi l'on commet de grossières erreurs. Dire cela, ce n'est pas dénigrer la technique, mais bien au contraire c'est la respecter assez pour ne pas la faire intervenir à contre-temps.

Les systèmes d'information ne sont pas des automates dont on attend qu'ils règlent tous les problèmes, mais des outils destinés à assister des opérateurs humains. La conception du système d'information doit donc considérer non le seul automate, mais le couple formé par l'automate et l'être humain organisé qu'il assiste.

2.4.3 Le semi-désordre

Si le système d'information est « parfait » l'entreprise peut paradoxalement devenir inefficace. Voici quelques exemples.

* *

Les défauts du système d'information obligent les opérateurs humains à faire chaque jour des interventions manuelles. Le jour où se produit un incident, ils savent comment faire car ils ont l'habitude de traiter les « pépins ». Si le système d'information était parfait il se produirait un incident tous les trois ans : les opérateurs perdraient l'habitude de réagir et feraient confiance au système. Le jour où se produirait un incident imprévu ils ne sauraient que faire. C'est pourquoi il est judicieux d'équiper une centrale nucléaire d'un logiciel moins complet qu'il ne pourrait l'être.

* *

La conception des avions est l'enjeu d'un conflit entre ingénieurs et pilotes. La qualité des avions étant élevée, les ingénieurs voient dans le « facteur humain » la cause résiduelle des accidents. Pour l'éliminer ils souhaitent concevoir l'avion « parfait » qui décollerait, volerait et se poserait sans pilote. Cependant les pilotes disent qu'il reste des situations où l'on a besoin du cerveau humain pour synthétiser, arbitrer et décider : l'avion doit comporter des automatismes, mais ceux-ci doivent assister le pilote et non le supplanter.

* *

Considérons une administration comme les impôts, la sécurité sociale ou l'ANPE. La réglementation évolue souvent, ce qui exige de modifier le système d'information. La modification est simple s'il s'agit de mettre à jour quelques paramètres, elle est complexe s'il faut redéfinir une partie d'un dossier et introduire des traitements nouveaux. Il faut de trois à six mois pour introduire une modification complexe dans le système d'information. Si celui-ci est de qualité médiocre, il faudra un an pour corriger les bogues provoquées par la modification. Pendant ce délai la réglementation aura encore changé.

Les agents se sont donc habitués à faire une partie de leur travail sur papier ou sur tableur, puis à saisir les données dans le système d'information. Cela comporte des inconvénients (erreurs de calcul ou de saisie, surcharge de travail, inefficacités diverses etc.), mais ce fonctionnement d'ensemble permet à l'administration d'être réactive et de mettre en œuvre sans délai une politique nouvelle.

* *

On a pu, dans certaines entreprises, modéliser la pratique professionnelle des agents pour automatiser leur démarche et gagner en rapidité. C'est ainsi que des banques ont conçu des systèmes experts de gestion de trésorerie. Cependant, si le contexte évolue, le système expert ne saura pas, lui, évoluer et il perdra en efficacité alors qu'un opérateur humain aurait adapté ses méthodes de travail et modifié ses « règles de pouce ». Il faut donc conserver, à côté du système expert qui fera le gros du travail, des opérateurs humains plus lents sans doute, mais dont le savoir pourra être périodiquement réinjecté dans le système expert pour le mettre à jour.

* *

La gestion d'un système d'information (ou d'un projet) navigue entre deux extrêmes. Suivre une méthodologie oblige à consacrer beaucoup de temps à la production de documents qui décrivent le système d'information sans faire nécessairement progresser son adéquation fonctionnelle. On peut aussi pratiquer l'artisanat « à l'ancienne » : dès qu'un métier a besoin de quelque chose, il demande aux informaticiens de le développer ; il revient à ceux-ci de gérer l'intendance, le métier ne se souciant pas des problèmes techniques du système d'information. Si l'on tolère la non-formalisation, les maîtrises d'ouvrage risquent de s'y engouffrer ; si on exige une formalisation complète, l'entreprise s'enlisera dans la production de documents en grande partie superflus. Le moyen terme efficace résulte du bon sens qui ne peut se formaliser entièrement.

* *

Un système d'information totalement désordonné n'est pas un système (la notion de système implique la cohérence) et ne contient d'ailleurs pas d'*information* : il stocke et produit des données qu'il est impossible de comparer et donc d'interpréter. Le désordre, c'est la mort du système d'information qui devient un *machin informe*.

La perfection serait une autre forme de mort : elle démobilise les opérateurs humains. Le mieux est ici l'ennemi du bien : il faut donc admettre une dose de « non qualité » (apparente) pour que la coopération entre l'automate et l'être humain soit le plus efficace possible.

Le laxisme peut prendre deux formes :

- le formalisme : il « fait sérieux », mais il aboutit soit à l'inefficacité, soit (conséquence moins dommageable) à la frustration du méthodologue qui ne parvient pas à se faire entendre ;

- le fatalisme, que traduisent l'expression « ça finira bien par tomber en marche » comme la phrase attribuée au président Queuille (1884-1970) : « il n'existe pas de problème dont une absence persévérante de solution ne finisse par venir à bout ».

Le semi-désordre est à l'opposé du laxisme : celui qui perçoit la façon dont l'APU et l'EHO s'articulent ne surestime pas les apports du formalisme et ne se laisse pas aller au fatalisme. La conception claire du résultat opérationnel à atteindre guide le choix de ses priorités et l'aide à définir les simplifications nécessaires.

Chapitre 3

Éclairage historique

« Do not be encumbered by history. Go off and do something wonderful »
(Robert Noyce, co-fondateur d’Intel ; source : www.intel.com/intel/)

« Une “révolution technologique” offre-t-elle l’occasion de remettre en cause les situations établies (...) ? Certains dirigeants français l’espéraient dès 1950, lorsque les premiers ordinateurs apparurent en Angleterre et aux États-Unis. Pourtant, les efforts accomplis (...) n’ont abouti qu’à faire de Bull, jadis quatrième constructeur mondial, une firme reléguée au treizième rang et dépendant des technologies américaines et japonaises [alors même que] la France réussissait à s’imposer dans (...) les télécommunications et le logiciel. Expliquer ce paradoxe conduit à analyser à la fois des caractères de longue durée : le purisme de l’école mathématique française, la rareté des compétences, la faiblesse de la recherche industrielle, la priorité des projets militaires sur les logiques de marché limitaient l’aptitude de la France à profiter de la “révolution informatique” (...); et les aspects contingents de cette histoire : échecs locaux et fautes stratégiques ont dépassé les étroites marges d’erreur permises (...) » (Pierre Mounier-Kuhn, [61]).

L’histoire de l’informatique intéresse un nombre croissant de chercheurs. Le champ ainsi ouvert à la réflexion est immense. Cette histoire ne peut en effet se comprendre que si l’on étudie et relie entre eux ses aspects logique, technique, philosophique, économique, sémantique, sociologique, industriel etc. ; il faut décrire à la fois ce qu’elle est, ce que ses promoteurs auraient voulu qu’elle fût, ce que ses utilisateurs en ont fait. Il faudra, pour venir à bout de ce programme, lui appliquer une attention aussi minutieuse que celle que Donald Knuth a consacrée aux algorithmes.

Parmi les chercheurs, on doit citer en France Pierre Mounier-Kuhn. Après avoir soutenu une thèse intitulée *L’informatique en France de la Deuxième guerre mondiale au Plan Calcul*, il anime un séminaire sur l’histoire de l’informatique à l’École pratique des hautes études. Un colloque international sur l’informatique et les réseaux s’est tenu à Grenoble en novembre 2002¹. En utilisant un moteur de recherche, on trouve sur la Toile de nombreux sites utiles dédiés à l’histoire de l’informatique.

1. Entre autres ouvrages, nous devons beaucoup à Philippe Breton [10] Robert Ligonnière [54], Paul E. Ceruzzi [15] et D. Sjobbema [77]

Paul E. Ceruzzi, *A History of Modern Computing*, MIT 1998

Paul Ceruzzi a tenté de construire une vue d'ensemble de l'informatique.

Il n'y arrive qu'en partie. D'une part, comme souvent en histoire, il s'arrête au seuil de l'époque actuelle : s'il décrit l'histoire de l'Internet, il ne décrit pas celle des langages orientés objet, ni des langages de modélisation, ni les « brokers » comme Corba, ni les changements apportés par l'Intranet dans les entreprises, ni même l'intelligence artificielle qui est pourtant une aventure déjà ancienne.

On souhaiterait par ailleurs qu'il abordât son sujet non seulement sur les plans technique et historique, mais aussi sur le plan philosophique. Comment exprimer autrement que dans le langage de la philosophie des projets visant à construire, relier et utiliser des concepts ? ce qui distingue tel langage de tel autre, ce qui fait la force ou la faiblesse d'outils bureautiques comme le tableur, la messagerie, la documentation électronique etc. ?

La lecture de ce livre est cependant utile. Il apporte des informations jusqu'ici inédites ou difficiles à trouver sur les origines des choix techniques, du vocabulaire, ainsi que sur les mécanismes de la réussite sur ce marché très concurrentiel.

La plupart des travaux érudits sont consacrés aux origines de l'informatique : certains partent des calculs gravés sur les tablettes mésopotamiennes, puis décrivent l'histoire des systèmes de numération (Ifrah [46]), les abaques des calculateurs de l'antiquité, la machine à calculer de Pascal (1623-1662), la machine programmable de Charles Babbage (1791-1871), enfin la machine mécanographique de Herman Hollerith (1860-1929).

Nous n'avons pas la prétention de rivaliser avec ces travaux spécialisés mais nous nous efforcerons de situer dans l'histoire les questions techniques, sociologiques ou économiques que pose aujourd'hui l'informatique. Cette démarche aide à délimiter ses concepts essentiels et à identifier les ressorts de sa dynamique. Il ne s'agit pas de collectionner des faits dont le souvenir inspirerait la nostalgie, ni de se repaître de l'image de machines désuètes, mais de prendre un peu de hauteur pour s'affranchir du poids du présent². L'analyse historique, prolongée jusqu'au passé récent, permet d'élucider le présent et d'éclairer le futur tout en laissant sa part à l'incertitude.

Le présent chapitre rassemble l'information sur trois thèmes historiques cruciaux : le micro-ordinateur ; les logiciels ; l'économie de l'informatique. Nous daterons les événements : la chronologie donne un point d'appui au raisonnement en aidant à placer les événements dans leur contexte ; elle aide aussi à identifier les innovations essentielles, celles qui, élargissant le domaine du possible, ont suscité d'autres innovations. Certaines dates servent ainsi de pivot à l'intelligence des faits.

2. Les travaux érudits, même quand ils se limitent à un empilage minutieux de documents, témoignages et références, ont pour but d'ouvrir la voie à la réflexion : l'érudition, tout aussi nécessaire que l'expérimentation, prépare comme elle son propre dépassement.

3.1 Histoire du micro-ordinateur

Nous avons décrit les circonstances de l'invention du microprocesseur chez Intel en 1971 (voir page ??). Un micro-ordinateur, c'est un ordinateur équipé d'un microprocesseur. Le micro-ordinateur ne pouvait donc pas naître avant le microprocesseur. Mais il ne suffisait pas que le micro-ordinateur naquît : il fallait aussi qu'il rencontrât un terrain favorable au succès commercial. Ce terrain a été préparé par les « hackers », ces pionniers qui ont rêvé le micro-ordinateur et la dissémination de l'informatique longtemps avant qu'ils ne fussent possibles, et qui s'étaient ainsi préparés à tirer parti de l'innovation technique dès que celle-ci se produirait.

* *

L'informatique était dans les années 60 l'affaire de professionnels stylés par IBM. Ostensiblement « sérieux », ils portaient costume, cravate noire et chemise blanche. Les utilisateurs n'étaient pas autorisés à approcher la machine.

Mais les « hackers » (Steven Levy [53]) (que l'on a appelé aussi « hobbyists ») revendiquaient le droit de comprendre comment la machine fonctionne, d'y accéder, de travailler en temps réel et donc de modifier la façon dont on utilisait l'ordinateur. Ils n'étaient pas des pirates, des briseurs de sécurité, des fabricants de virus : le mot « hacker » n'a pris ce sens péjoratif que récemment. Les hackers des années 60 et 70 étaient des *pionniers*. Avant eux, l'ordinateur était une grosse machine sans écran, sans carte sonore, sans traitement de texte, sans tableur, sans réseau etc. Ils ont inventé les procédés qui ont permis d'introduire tous ces perfectionnements.

Un « good hack », c'était une astuce qui marche, un procédé qui permet de faire quelque chose de nouveau par l'automate - peu importe d'ailleurs si l'on ne sait pas exactement pourquoi ni comment cela fonctionne. La passion des hackers était de créer de « good hacks », non de les vendre ni de « faire du business ». Ils copiaient sans vergogne les programmes, les modifiaient, les communiquaient. L'« open source » (ou « logiciel ouvert »), dont le programme source est fourni et modifiable à volonté, allait de soi ainsi que la gratuité.

La première équipe de hackers, la plus flamboyante, fut celle du MIT ; puis d'autres équipes de passionnés se sont créées ailleurs : aux Bell Labs d'AT&T, à l'université de Berkeley et, dans les années 70, au Palo Alto Research Center (PARC) de Xerox. Les hackers du MIT estimaient avoir le droit et même le devoir d'accéder à la machine quelles que soient les interdictions et réglementations, et même s'il fallait fabriquer de fausses clés pour se faufiler de nuit dans un centre informatique.

Ils ont mis au point des langages et des méthodes, inventé l'intelligence artificielle etc. Le système d'exploitation Unix (1969), le langage C (1972) mis au point aux Bell Labs ont été d'abord distribués gratuitement ou pour un paiement symbolique, ce qui a permis de les perfectionner rapidement. La souris a été inventée par Douglas Engelbart en 1968, les fenêtres et menus déroulants par Dan Ingals au PARC en 1974. Enfin les hackers ont cherché à mettre l'ordinateur à la disposition de tout le monde, d'abord en disséminant des terminaux, puis en mettant au point le micro-ordinateur.

Steven Levy a décrit les rêves et les ambitions de ces passionnés qui travaillaient parfois trente heures d'affilée et sacrifiaient leur santé et leur vie affective à l'exploration des possibilités de l'ordinateur, au développement d'outils qui permettraient de

les concrétiser. Leur imprégnation par le langage informatique les coupait des modes de communications naturels et les isolait des autres êtres humains. Ils avaient une morale, « l'éthique des hackers » (voir page 87).

L'émergence du commerce des logiciels au milieu des années 70 a mis un terme à cette époque qui se prolonge cependant dans l'école du logiciel ouvert.

* *

Les hackers ont travaillé sur ce qui était à leur époque la « couche critique » de l'informatique, cette expression désignant l'ensemble des questions qui se trouvent sur le front de taille de la discipline.

Il s'agissait dans les années 60 de mettre au point les langages qui permettraient de diversifier les utilisations de l'ordinateur : la couche critique était alors celle du *logiciel*. Dans les années 70 il s'est agi de mettre l'ordinateur dans les mains de chacun en s'appuyant sur les nouveaux micro-processeurs : la couche critique fut alors celle du *matériel* ; une fois traitée, il a été possible de réaliser les développements qui ont conduit au micro-ordinateur en réseau d'aujourd'hui, avec ses interfaces graphiques, son équipement multimédia et l'Internet.

La couche critique est aujourd'hui celle de *l'utilisation* collective, organisée, de l'informatique par les entreprises, la société et les individus. L'utilisation individuelle pose des questions qui relèvent de la psychologie ; l'utilisation collective pose des questions qui relèvent de la sociologie et de l'organisation. Sociologie et psychologie constituent deux couches, différentes mais solidaires, du même empilement. Les héritiers des « hackers », des pionniers des années 60 et 70, sont ainsi aujourd'hui les personnes qui travaillent sur le langage de l'entreprise, l'organisation transverse, l'articulation du système d'information avec la stratégie, la modélisation des processus, l'urbanisation des systèmes d'information etc.

Reprenons le chemin qui a conduit du premier micro-ordinateur aux réseaux de machines d'aujourd'hui.

Un précurseur : le Kenbak-1 (1971)

John V. Blankenbaker présente le Kenbak-1 au printemps de 1971 ; cette machine, dotée d'une RAM de 256 octets, était vendue en kit au prix de 750 \$. Comme le 4004 d'Intel n'était pas encore commercialisé - il ne sera mis sur le marché qu'en novembre 1971 - l'unité centrale du Kenbak-1 comporte plusieurs composants MSI et LSI : il ne s'agit donc pas d'un micro-ordinateur au sens exact du terme, mais d'un ordinateur à circuits intégrés. Le Kenbak-1 avait pour but d'aider l'apprentissage de la programmation. Après en avoir vendu 40 exemplaires, la Kenbak Corporation a été supprimée en 1973.

Le premier micro-ordinateur

En 1971 André Thi Truong, ingénieur français d'origine vietnamienne, fonde la société R2E (Réalisations Études Électroniques). En 1972 l'INRA (Institut national de la recherche agronomique) commande à R2E un système informatique transpor-

« The Hackers Ethic »

(Steven Levy [53] p. 38)

1) *L'accès aux ordinateurs - et à tout ce qui peut vous apprendre quelque chose sur la façon dont le monde fonctionne - doit être illimité. Respectez le « Hands-On Imperative » !* Quand on rencontre un obstacle, il faut « y mettre les mains » et résoudre le problème sans attendre que l'on vous y invite ou que l'on vous y autorise ^a.

2) *L'information doit être gratuite.* La gratuité de l'information et des logiciels est supposée plus efficace, par les synergies qu'elle permet, qu'une économie où les logiciels seraient vendus sur le marché et protégés par des copyrights.

La transition entre la gratuité et le marché sera le « software flap » provoqué par Bill Gates lorsqu'il publia en février 1976 sa « Open Letter to Hobbyists » dans *Computer Notes*, « newsletter » des utilisateurs de l'Altair.

3) *Ne pas faire confiance à la hiérarchie, promouvoir la décentralisation.* Les hackers étaient, bien avant d'autres, partisans de l'organisation transverse qu'ils jugeaient seule efficace. Ils étaient par ailleurs insensibles aux prestiges de la hiérarchie, comme le montre la règle suivante :

4) *Juger les hackers selon la qualité de leurs hacks et non selon des critères farfelus comme le diplôme, l'âge, la race ou le grade.* Seule compte la compétence, le niveau atteint dans la maîtrise de la machine : il s'agit de contourner les obstacles que celle-ci oppose à ceux qui veulent la plier à leurs besoins.

5) *Vous pouvez créer de l'art et de la beauté avec un ordinateur.* L'ordinateur n'est pas seulement fait pour calculer, comme le suggère le mot « computer », ni pour mettre de l'ordre, comme suggère le mot « ordinateur » : on doit pouvoir l'utiliser pour faire de la musique, dessiner, créer des mondes imaginaires qui donneront aux rêves un prolongement (presque) aussi vrai que le monde réel.

C'est l'origine des mondes virtuels qui ont eu tant d'importance lors de l'explosion du marché des jeux. Il est significatif qu'une bonne part du progrès des ordinateurs en performance, convivialité et ergonomie provienne de leur utilisation ludique. Il est intéressant de noter que les « hackers de la troisième génération », qui ont grandi dans les années 70, sont venus à l'informatique par la pratique des jeux (Bennahum [4]).

6) *Les ordinateurs peuvent améliorer votre vie.* Les hackers n'ignorent pas les risques que comportent certains usages de l'ordinateur (ses utilisations militaires les effraient), mais ils affirment que des utilisations utiles, progressistes, constructives sont possibles.

^a C'est la même attitude que résume le slogan de Nike : « Just do it ».

FIG. 3.1 – *Le Micral (1973)*

table. R2E conçoit alors un ordinateur fondé sur le processeur 8 bits Intel 8008³. Le Micral est créé en six mois. La machine a été conçue par André Thi Truong, le logiciel par Philippe Kahn (figure 3.1).

Gros comme l'unité centrale d'un PC actuel, le Micral ne dispose ni d'écran ni de clavier. La saisie se fait en binaire, en manipulant des interrupteurs. On peut lui connecter un lecteur de bandes perforées. 500 Micrals sont produits la première année ; ils sont vendus 8 450 F pièce (1 750 \$ de l'époque). En juin 1973 la revue américaine *Byte* invente le mot « microcomputer » pour baptiser le Micral : le micro-ordinateur est né.

Le Micral n'a pas connu le succès commercial ; il a été utilisé essentiellement pour automatiser des postes de péage des autoroutes⁴.

Autres précurseurs

Le Scelbi (« Scientific, Electronic and Biological ») de la Scelbi Consulting Company (Milford, Connecticut) fut annoncé dans l'édition de mars 1974 de la revue *QST*. Construit autour du microprocesseur 8008 d'Intel, le Scelbi avait 1 Ko de mémoire programmable et il était vendu 565 \$. 15 Ko de mémoire supplémentaire étaient fournis pour 2760 \$.

Le Mark-8, conçu par Jonathan Titus, était lui aussi construit autour de l'Intel 8008. Il fut annoncé dans le numéro de juillet 1974 de *Radio Electronics*.

L'Altair 8800 (1974)

Intel sortit le microprocesseur 8080 en juillet 1974. Il était vingt fois plus rapide que le 4004 et son prix pouvait descendre jusqu'à 3 \$ pour des commandes en quantité. Ce microprocesseur sera au cœur de la conception de l'Altair, la machine qui a enfin fait démarrer le marché du micro-ordinateur.

MITs (« Micro Instrumentation Telemetry System »), petite entreprise d'Albuquerque au Nouveau Mexique dirigée par Ed Roberts, a lancé l'Altair 8800 en décembre 1974. La photographie de ce micro-ordinateur paraît pour la première fois

3. Sorti en avril 1972, le 8008 comportait 3500 transistors. Sa vitesse était de 108 kHz, sa mémoire adressable de 16 Ko.

4. En 1978, R2E est absorbée par Bull. En 1982, la filiale américaine de Bull concevra son premier compatible PC, le Bull Micral. Truong quittera Bull en 1983.

sur la couverture du numéro de janvier 1975 de la revue *Popular Electronics* (figure 3.2). Dans les deux mois qui suivent des milliers de commandes arrivent à MITS.

L'Altair était commercialisé en kit et il fallait beaucoup d'habileté pour le monter. L'unité de base avait 256 octets de RAM et coûtait 395 \$: ainsi l'Altair était beaucoup moins cher que le Micral. Comme il n'existait pas de logiciel, les utilisateurs devaient écrire leurs propres programmes puis les saisir en binaire en appuyant sur des boutons⁵. Le résultat était affiché en binaire sur des diodes.

On pouvait ajouter au système un bus et des cartes d'extension. En quelques mois les cartes qui permettaient d'accroître la mémoire ou de raccorder des terminaux furent disponibles. Le télétype ASR-33 fournit un clavier, une imprimante et un support de stockage sur bande de papier perforé. MITS commercialisa par la suite d'autres versions améliorées (8800a, 8800b).

L'Altair et la naissance de Microsoft

Les origines de Microsoft sont intimement liées à l'arrivée de l'Altair sur le marché. Suivons la chronologie telle qu'elle est décrite dans la biographie de Bill Gates⁶.

Paul Allen (né en 1953, alors employé par Honeywell) et Bill Gates (né le 28 octobre 1955, alors étudiant en deuxième année à Harvard) étaient des amis passionnés par l'informatique, des hackers. L'article de *Popular Electronics* les incita à programmer un interpréteur Basic pour l'Altair : ce sera le premier langage de programmation pour micro-ordinateur. Allen et Gates vendirent la licence de cet interpréteur à MITS le 1er février 1975.

Le 1er mars 1975, Allen est embauché par MITS comme « Director of Software ». Le 7 avril 1975, *Computer Notes*, la « newsletter » des utilisateurs de l'Altair, annonce que l'interpréteur Basic est opérationnel⁷. Une version 2.0 est publiée le 1er juillet 1975. Le 22 juillet 1975, Allen et Gates signent avec MITS un accord de licence. Le 29 novembre 1975, Gates utilise dans une lettre à Allen le nom « Micro-Soft » : c'est la première mention écrite connue de ce nom.

Le 3 février 1976, Bill Gates publie dans *Computer Notes* sa « Open Letter to Hobbyists » [36] (voir ci-dessous). Le 1er novembre 1976, Allen quitte MITS pour rejoindre Microsoft à plein temps.

Le 26 novembre 1976, la marque « Microsoft » est enregistrée auprès du bureau du secrétariat d'état du Nouveau Mexique pour « nommer des programmes informatiques, des systèmes d'exploitation et des services ». La déclaration mentionne que le nom de Microsoft a été utilisé depuis le 12 novembre 1975.

La lettre ouverte de Bill Gates aux « Hobbyistes »

Dans sa lettre ouverte (voir le texte en anglais page 91 et la traduction en français page 92), Bill Gates accuse de vol les « hobbyistes » qui ont piraté son interpréteur Basic. Il leur reproche d'empêcher la production de bons logiciels et termine par une phrase prophétique : « Rien ne me ferait plus plaisir que de pouvoir embaucher dix programmeurs pour inonder le marché de bons logiciels ».

5. Des cals se formaient rapidement au bout de leurs doigts...

6. Source : www.microsoft.com/billgates/bio.htm

7. « Up and running ».

Cette lettre est importante pour l'histoire de l'informatique : elle inaugure le modèle économique du « logiciel compilé marchand vendu en boîte » qui succédera au modèle du « programme source librement retouchable » des hackers. Bill Gates avait vingt ans lors de sa publication.

Même si le modèle « open source », revitalisé par l'Internet, est revenu en force vingt ans après la publication de cet article, le modèle marchand a eu son utilité : il était sans doute le seul dans les années 70 qui pût permettre la production rapide des logiciels nécessaires au succès du micro-ordinateur. Accessoirement, il fut aussi à l'origine de la croissance de Microsoft et de la fortune de Bill Gates.

3.2 Les apports du PARC de Xerox

Les dirigeants de Xerox, entreprise qui avait construit sa fortune sur le marché des photocopieurs avec la « xérographie », ont eu l'intuition du caractère porteur de la microélectronique et de l'informatique. Xerox avait acheté Scientific Data Systems en 1969⁸ ; elle décida alors de créer un centre de recherche.

Le « Palo Alto Research Center » (PARC) de Xerox est créé en 1970. Il sera situé dans la Silicon Valley, tout près de l'Université de Stanford, afin de pouvoir bénéficier des apports de la recherche universitaire et de la force de travail des étudiants. Quelques-uns des hackers les plus créatifs se rassemblent au PARC. Ils vont apporter à l'informatique des innovations dont le micro-ordinateur tirera parti (voir la liste des principales d'entre elles page 93).

L'histoire du PARC comporte cependant un paradoxe : si Xerox a généreusement financé ce centre de recherche, elle n'a pratiquement pas tiré parti de ses découvertes qui toutes seront finalement commercialisées par d'autres entreprises (Hiltzik [43]). Xerox a construit l'Alto, mais c'est IBM qui a lancé le PC en 1981. Xerox a inventé l'interface graphique avec fenêtres, souris et menus déroulants, mais c'est Apple qui a lancé le Macintosh et Microsoft qui a produit Windows. Xerox a conçu le traitement de texte Wysiwyg⁹, mais c'est Microsoft qui a produit Word. Xerox a mis au point le protocole de réseau Ethernet, mais aujourd'hui le marché des réseaux locaux se partage entre Cisco et 3Com. Xerox a inventé l'imprimante à laser, mais il s'est fait précéder sur ce marché par IBM en 1975.

L'une des dates les plus importantes de l'histoire de l'informatique est celle de la visite de Steve Jobs et d'une équipe d'ingénieurs d'Apple¹⁰ au PARC en décembre 1979. La démonstration à laquelle ils assisteront les incitera à introduire l'interface

8. L'année même où l'Arpanet, précurseur de l'Internet, devient opérationnel (voir page ??).

9. Le Wysiwyg (« What You See Is What You Get ») est le principe le plus important de l'interface graphique : chaque manipulation de l'image sur l'écran entraîne une modification prévisible de l'état du système. Les éléments de cette métaphore sont les fenêtres (Windows), menus, icônes, boutons, onglets ainsi que le pointeur. Les fenêtres permettent la représentation simultanée de plusieurs activités sur l'écran. Les menus permettent de choisir les prochaines actions. Les icônes, boutons, onglets etc. attribuent une forme concrète aux objets informatiques. L'outil de pointage, souris ou track-ball, sélectionne fenêtres, menus, icônes etc.

10. Apple Computer avait été créé en avril 1976 par Steven Wozniak et Steven Jobs. Leur Apple I (1976) n'a pas été pris au sérieux par les hobbyistes, mais l'Apple II (1977), premier ordinateur personnel carrossé en plastique et présentant une interface en couleur, a connu le succès. Le premier tableur, Visicalc (1979), a été réalisé pour l'Apple II. L'Apple III sortira en 1980.

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds \$40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than \$2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guy who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates
General Partner, Micro-Soft

Traduction de la lettre de Bill Gates :

3 février 1976

Lettre ouverte aux « Hobbyistes »

De mon point de vue, le manque de bon cours de programmation, de bons livres et de logiciels est aujourd'hui pour les hobbyistes la question la plus critique. Sans un bon logiciel, si l'utilisateur ne connaît pas la programmation, l'ordinateur du hobbyiste reste stérile. Écrira-t-on du logiciel de qualité pour ce marché?

Voici près d'un an, Paul Allen et moi avons anticipé sa croissance, embauché Monte Davidoff et développé le BASIC pour l'Altair. Le travail initial n'a pris que deux mois mais nous avons tous les trois dû passer l'essentiel de l'année dernière à documenter ce BASIC, l'améliorer et l'enrichir en fonctionnalités. Nous avons maintenant des versions BASIC pour les mémoires 4K, 8K, étendues, ROM et disque dur. Nous avons utilisé pour plus de 40 000 dollars de temps d'ordinateur. Nous avons reçu des réactions positives de centaines de personnes qui utilisent ce BASIC. Cependant, deux faits surprenants apparaissent : 1) la plupart de ces « utilisateurs » n'ont jamais acheté le BASIC (il a été acheté par moins de 10 % des possesseurs d'un Altair), 2) le montant des redevances venant de nos ventes aux hobbyistes rémunère le travail fourni sur le BASIC de l'Altair à moins de 2 dollars de l'heure.

Pourquoi cela? La plupart des hobbyistes savent bien qu'ils volent le logiciel. Il faut bien acheter le matériel, mais le logiciel, ça se partage. Qui se soucie de rémunérer les gens qui ont travaillé pour le produire?

Est-ce honnête? Si vous volez le logiciel, vous ne pourrez pas vous retourner contre MITS en cas de problème. MITS ne fait aucun profit en vendant le logiciel. La redevance qui nous est payée, le manuel, la bande etc. en font une opération tout juste équilibrée. Ce que vous faites, c'est d'empêcher la production de bon logiciel. Qui peut se permettre de faire travailler des professionnels pour rien? Quel hobbyiste pourrait mettre trois hommes*année dans la programmation, détecter toutes les bogues, documenter le produit puis le distribuer pour rien? Le fait est que personne, en dehors de nous, n'a investi d'argent dans le logiciel pour les hobbyistes. Nous avons écrit le BASIC pour le 8080, nous sommes en train d'écrire l'APL pour le 8080 et le 6800, mais rien ne nous incite à mettre ces logiciels à la disposition des hobbyistes. Vous êtes tout simplement des voleurs.

Que penser de ceux qui revendent le BASIC pour Altair? Ne se font-ils pas de l'argent sur le marché des logiciels pour hobbyistes? Oui, mais ceux que l'on nous a signalés pourraient finalement y perdre. Ils donnent une mauvaise réputation à tous les hobbyistes : on devrait les chasser des réunions des clubs où ils apparaissent.

J'aimerais recevoir des lettres de tous ceux qui souhaitent payer leur dette envers nous, ainsi que de ceux qui ont des suggestions ou des commentaires à formuler. Écrivez-moi à 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Rien ne me plairait davantage que de pouvoir embaucher dix programmeurs pour inonder de bons logiciels le marché des hobbyistes.

Bill Gates

Directeur Général, Micro-Soft

graphique dans le Lisa (1983) et dans le Macintosh (1984).

Les inventions du PARC

1971 : Alan Kay et son équipe mettent au point la première version de Smalltalk, langage de programmation orienté objet qui influencera C++ et Java ; Gary Starkweather met au point la première imprimante à laser.

1973 : Chuck Thacker, Butler Lampson et Alan Kay mettent au point l'Alto, ordinateur qui ne sera jamais commercialisé. Bob Metcalfe crée le protocole de réseau local Ethernet.

1974 : Dan Ingals invente un système qui offre l'interface avec fenêtres superposées et menus déroulants. Charles Simonyi, Tim Mott et Larry Tesler produisent le premier traitement de texte convivial.

1979 : James Clark conçoit le microprocesseur graphique 3D qui fera par la suite la fortune de Silicon Graphics.

30 septembre 1980 : les spécifications du réseau Ethernet sont publiées. C'est la première fois qu'une invention du PARC est commercialisée.

27 avril 1981 : Xerox lance la station de travail Star (16 000 \$), descendant commercial de l'Alto et des travaux du PARC. Mais le 24 août 1981 IBM lance le PC (2 000 \$) qui rend instantanément le Star obsolète.

1983 : Après la démission forcée de Bob Taylor le 19 septembre 1983, plusieurs des ingénieurs du PARC démissionnent. Le style de travail du PARC ne sera plus le même.

Janvier 1984 : Apple lance le Macintosh, incarnation réussie de l'ordinateur personnel conçu par le PARC.

Pourquoi Xerox n'a-t-il pas utilisé les travaux du PARC ? Il est tentant mais trop facile d'expliquer cela par des comportements personnels, des conflits politiques dans l'entreprise ou la stupidité de bureaucrates incapables de percevoir le potentiel d'une innovation. Ces facteurs ont joué, mais ce ne sont pas les plus importants. En fait il n'était pas facile de réussir dans le domaine des ordinateurs personnels, comme l'ont montré les échecs rencontrés ensuite par IBM et même par Apple ; et s'il était en 1981 possible pour Apple (40 personnes) de prendre de gros risques, c'était pratiquement impossible pour Xerox (125 000 personnes).

Xerox s'était spécialisée sur le marché des photocopieurs, grosses machines que l'on installe dans les secrétariats et que le fournisseur fait rémunérer à la copie. Le succès de la xérogaphie avait mis Xerox en situation de monopole et l'avait convaincue qu'il suffisait d'offrir de bons produits, mûrement conçus, pour que les clients se jettent dessus. Cela ne la préparait pas au marché de l'informatique personnelle où les acheteurs sont les directeurs informatiques, où il n'est pas question de facturer à la consommation et où la concurrence contraint à se battre pour chaque pourcentage de part de marché.

L'intérêt de Xerox pour l'innovation était sincère, mais affaire de principe plus que de stratégie ; pour que Xerox puisse devenir un acteur sur ce marché, il aurait fallu que ce fût une entreprise nouvelle, sans histoire, sans habitudes, sans organisation, libre de se modifier comme le firent Apple et Microsoft.

Toute grande entreprise, toute organisation structurée par l'histoire répugne à changer et a donc tendance à refuser l'innovation. Cela explique en partie les échecs de l'informatique française : celle-ci a été poussée par une administration que l'on peut considérer comme la plus grande entreprise du monde et qui, comme toute grande entreprise, est corsetée par son histoire.

Pour concevoir le PC, IBM a dû créer en son sein une organisation indépendante que le président a protégée contre le reste de l'entreprise, et cela n'a pas encore suffi : la culture d'entreprise ayant rejeté cette greffe, IBM n'a pas pu profiter du PC pour dominer le marché de la micro-informatique. Ce sont Intel et Microsoft, structures minuscules en 1981, qui ont raflé la mise.

3.3 IBM et le PC

La relation entre IBM et le PC se déroule selon un déterminisme aussi implacable que celui d'une tragédie grecque : une entreprise dont la capitalisation boursière dépassait celle de l'ensemble des entreprises allemandes, dont la croissance avait apporté le bien-être à des vallées entières près de New-York et de Boston, fière de ses traditions de qualité et de sérieux, trébuche sur le micro-ordinateur en raison même de ces traditions et manque de s'effondrer (Carroll [14]).

En 1980, le micro-ordinateur est un défi pour IBM. Le PET (Personal Electronic Transactor) de Commodore, le TRS-80 de Radio Shack et l'Apple II, sortis tous les trois en 1977, séduisent des clients. Frank Cary, président d'IBM, estime que son entreprise doit être présente sur ce marché. Le micro-ordinateur étant une petite machine, sa réalisation ne devrait pas poser de problème à la plus grande entreprise informatique du monde !

Cependant IBM, habituée aux gros systèmes, n'arrive pas à s'intéresser au micro-ordinateur. Le 4 juillet 1980, Cary pique un coup de sang et décide de s'occuper directement du problème. Le micro-ordinateur d'IBM sera réalisé par une équipe installée à Boca-Raton, en Floride, et séparée du reste de l'entreprise. Elle rapportera directement au président qui la financera lui-même et la protégera.

Le président veut son micro-ordinateur : il l'aura. IBM sort son premier PC (« Personal Computer ») en août 1981. C'est une réussite technique et le succès commercial est immédiat. Certains clients étaient en effet restés réticents devant le micro-ordinateur qu'ils croyaient peu sérieux. Ces réticences tombent lorsque IBM lui-même, temple du sérieux, cautionne le micro-ordinateur en produisant le PC. Il s'ensuit une avalanche de commandes.

Pour sortir le PC rapidement IBM avait sous-traité des parties essentielles à d'autres entreprises : le microprocesseur à Intel, le système d'exploitation MS-DOS à Microsoft¹¹. Le ver était ainsi dans le fruit. Le micro-ordinateur étant étranger

11. La coopération avec Microsoft fit apparaître quelques défauts chez IBM : « Les gens de Microsoft se plaignaient de la méthode de programmation d'IBM [...] IBM mesurait le nombre de lignes produites par chaque programmeur, ce qui encourageait la production de code inefficace. Les gestionnaires d'IBM se plaignaient parce que, selon leur système de mesure, Microsoft ne faisait pas sa part du travail. Ils disaient que si l'on comptait le nombre de lignes, Microsoft faisait en fait un travail négatif, ce qui signifiait que Microsoft aurait dû payer IBM parce qu'il condensait le code » Cette façon de mesurer la production pousse à écrire de programmes lourds. IBM la jugeait efficace parce qu'elle induisait une forte utilisation des machines et incitait à terme les clients à acheter de

à sa culture, IBM a manqué de flair et pris une disposition catastrophique au plan stratégique : il n'a pas introduit de clause d'exclusivité dans les contrats avec Microsoft et Intel. Cela permettra à des fabricants de s'approvisionner auprès de ces entreprises et de produire à partir de 1986 des micro-ordinateurs « compatibles PC », des « clones », puis d'engager contre IBM une guerre des prix qui le contraindra à réduire sa marge.

Intel et Microsoft, en position de monopole sur un marché en croissance rapide, feront fortune. Le marché des mainframes, qui procurait à IBM l'essentiel de son profit, souffrira de l'arrivée des PC. Les PC dégageront peu de marge en raison de la pression de la concurrence. Les années 80 seront alors pour IBM une descente aux enfers : baisse des résultats et du cours de l'action, compression des effectifs etc.

Certains, comme Paul Carroll dans *Big Blue* [14], ont relevé avec une ironie mordante les dysfonctionnements d'IBM ; d'autres, comme Gérard Dréan [21] expliquent les difficultés qu'a rencontrées IBM par l'intensité des changements que nous venons de décrire. Il est certain que mieux une entreprise était adaptée à l'informatique du début des années 80, plus il lui était difficile de s'adapter à l'informatique de la fin des années 80 (voir page 104) et d'autant plus qu'elle était plus grosse, la complexité et la rigidité de l'organisation étant fonction croissante de la taille de l'entreprise.

Il est futile d'ironiser sur les malheurs des grandes entreprises : le fait est qu'IBM, que l'on a pu croire presque morte en 1993, s'est redressée par la suite et s'est réorientée vers le secteur des services, ce qui prouve qu'elle n'avait pas perdu tout son potentiel.

3.4 Chronologie du PC

Voici les dates des principaux événements qui ont marqué la suite de l'histoire du micro-ordinateur :

1982 : Compaq commercialise le premier micro-ordinateur portable (il pèse 15 kg !).

1983 : Apple lance le Lisa, premier ordinateur ayant une interface graphique (menus déroulants, fenêtres, corbeille etc.) : le Lisa a été inspiré par les travaux du PARC.

La norme IEEE 802.3 pour les réseaux locaux Ethernet est publiée. Elle concrétise les spécifications produites par le PARC en 1980. C'est le début de la généralisation des réseaux locaux dans les entreprises.

1984 : Lotus sort le tableur Lotus 1-2-3 (voir page 117 l'histoire du tableur) qui incitera les entreprises à acheter des PC. Apple commercialise le Macintosh, qui apparaîtra comme le grand concurrent du PC : les utilisateurs se partagent entre partisans de l'une ou l'autre des deux machines.

1985 : IBM lance le PC AT qui a un grand succès. IBM annonce en octobre le réseau Token Ring, qui concurrencera Ethernet dans les entreprises industrielles (voir page ??). Intel lance le même mois le processeur 80386 à 16 MHz qui améliore de façon significative la puissance du PC. Microsoft livre en novembre Windows 1.0 qui apporte l'interface graphique aux utilisateurs de PC.

1986 : Les bases de données sur PC se développent avec dBASE d'Ashton et

nouveaux ordinateurs plus puissants (Paul Carroll [14] p. 101).

Tate. Microsoft lance le tableur Excel (d'abord connu sous le nom de Multiplan). DEC connaît sa meilleure année, mais ce sont les derniers feux du mini-ordinateur, dont le marché est coincé entre la gamme des mainframes et le micro-ordinateur. Compaq lance le marché des clones PC en produisant le premier PC 386. L'utilisation de la messagerie électronique se développe aux États-Unis : MCI et Compuserve offrent un lien entre leurs messageries respectives.

1987 : les PC 386 détrônent les PC AT. IBM lance la série PS/2 et le système d'exploitation OS/2. Apple lance le Macintosh II. Le 12 juillet 1987, Robert Solow formule dans la *New York Review of Books* son célèbre paradoxe : « You can see the computer age everywhere but in the productivity statistics. »

1988 : Unix gagne en notoriété et érode la confiance dans les mini-ordinateurs et les mainframes. Apple poursuit Microsoft et HP en justice à propos de l'interface du PC. Compaq prend la tête d'un consortium de fournisseurs connu comme « gang des neuf », et crée le standard EISA pour contrer le Micro Channel du PS/2 d'IBM. Il annonce le mini-ordinateur AS/400.

1989 : Ethernet 10BaseT démarre. C'est l'année des réseaux locaux de PC ; les hubs et adaptateurs de SynOptics et 3Com ont un grand succès. Cela prépare la voie des applications client/serveur des années 90. En avril Intel annonce le processeur 486. IBM sort OfficeVision, sa première suite d'applications. Apple lance un Macintosh portable (il pèse 7 kg!).

1990 : C'est l'année des routeurs et des WAN (« Wide Area Network ») interconnectant les réseaux locaux de l'entreprise. Microsoft lance Windows 3.0. En septembre, IBM et Microsoft redéfinissent leur partenariat : IBM prend la responsabilité d'OS/1.x et 2.x, et Microsoft celle de l'OS/2 portable, de DOS et de Windows. Motorola lance le processeur 68040, Apple lance des Macs bas de gamme : Classic, LC et IIsi.

1991 : Windows est en position de monopole et OS/2 disparaît de la scène. Naissance du World Wide Web : Tim Berners-Lee, au CERN de Genève (« Conseil Européen pour la Recherche Nucléaire »), monte le premier serveur Web (voir page ??). Les dépenses des entreprises en informatique dépassent pour la première fois les dépenses en équipement industriel, agricole et en construction. Apple lance la première génération de Powerbooks.

1992 : C'est l'année des applications sur réseau local et aussi celle du groupware avec Lotus Notes. L'outsourcing émerge avec un gros contrat passé par Kodak. IBM et Microsoft mettent fin à leur accord de coopération. Pour la première fois, les comptes annuels d'IBM font apparaître une perte de 564 millions de dollars. IBM lance le premier de ses Notebooks : le ThinkPad.

1993 : Début du déploiement du Pentium. Les pertes d'IBM sont les pires de son histoire : 4,97 milliards de dollars pour un chiffre d'affaires de 64,5 milliards. Lou Gerstner remplace John Akers à la tête d'IBM : c'est le premier « outsider » qui prenne le poste de CEO. Apple perd son procès contre Microsoft et HP. Intel lance le processeur 60 MHz Pentium, Apple sort le Newton, Novell annonce NetWare 4.0, Lotus Notes 3.0 démarre, Microsoft lance Windows NT.

1994 : L'architecture client/serveur devient à la mode. L'erreur de calcul révélée sur la puce Pentium suscite un drame de relations publiques pour Intel, qui y met un terme en remplaçant toutes les puces. Microsoft annonce que Windows 95 ne sera pas livré avant août 1995, ce qui mécontente beaucoup d'utilisateurs contraints

à revoir leurs plans de migration vers le 32 bits. La frénésie des fusions et acquisitions continue : Novell achète WordPerfect (voir page 113 l'histoire du traitement de texte), Aldus et Adobe fusionnent. Apple entre sur le marché de la vente en ligne avec eWorld. Netscape, le futur chéri de Wall Street, fait ses débuts.

1995 : Les Notebooks deviennent une alternative au desktop avec les portables Pentium. Il en résulte un développement du télétravail. IBM fait une offre de 3,5 milliards de dollars pour acheter Lotus. En août, Microsoft livre Windows 95 et Intel lance le Pentium Pro à 150-200 MHz. Compuserve, AOL et Prodigy commencent à offrir des accès au Web, et Netscape lance la troisième plus importante augmentation de capital sur le Nasdaq.

1996 : Windows 95 confirme son emprise sur le PC. Windows NT gagne du terrain contre NetWare comme plate-forme pour serveur. Les Network Computers se concrétisent par de vrais produits. Les Intranet d'entreprise se répandent. Java gagne en notoriété. Les entreprises commencent à développer des sites Web. Microsoft adopte enfin le Web.

1997 : C'est l'année de l'Intranet. Le commerce électronique démarre. La navigation sur le Web devient facile avec des browsers et des outils de recherche améliorés. La puissance de traitement s'accroît lorsque Intel annonce le Pentium 200 MHz avec la technologie MMX.

1998 : La perspective de l'an 2000 effraie tout le monde. Le manque de personnel en informatique devient aigu. L'outsourcing et les services s'épanouissent. Le grand thème à la mode est le commerce sur l'Internet. Plusieurs événements importants non reliés au Web se produisent : achat de Digital par Compaq, durcissement de la bataille entre Microsoft et le ministère de la justice.

3.5 Innovations importantes

Le CD-ROM

Les CD-Rom ont été initialement conçus pour porter 74 minutes de son de haute qualité. Transformés en supports de données (650 Mo), ils épargnent aux utilisateurs les heures de travail qu'ils devaient consacrer à charger une application disquette après disquette.

Les CD-ROM ont pratiquement remplacé les disquettes, d'autant plus qu'il devient possible de graver des CD. Cependant le CD-ROM est concurrencé par les DVD-ROM et par les disques optiques capables de contenir 4,7 Go.

Le chiffrement

Sans les algorithmes de chiffrement, le commerce électronique n'aurait pas pu se développer sur l'Internet : le chiffrement est crucial pour identifier les parties impliquées dans une transaction et sécuriser les transactions.

PGP (« Pretty Good Privacy »), créé par Philip Zimmermann, a suscité une accélération avec son architecture ouverte et ses techniques mathématiques robustes. Les autres leaders dans ce domaine sont RSA Data Security, qui teste les limites des techniques de chiffrement, et Certicom, avec des innovations comme la courbe de protection elliptique (économe en largeur de bande) qui fait partie de l'offre PalmNet

de 3Com.

Ethernet et les réseaux locaux de PC

Plus que tout autre produit, NetWare 2.11, Ethernet 10BaseT et 10Base2 ont apporté aux entreprises le partage des données et de la puissance de traitement. NetWare a transformé le micro-ordinateur en une machine analogue à un mainframe, Ethernet a permis de relier les ordinateurs entre eux. Sans la synergie entre ces technologies, l'explosion de l'Internet n'aurait pas pu se produire car le réseau d'ordinateurs serait resté coûteux. Elles sont évolutives : Ethernet devient sans cesse plus rapide et s'étend aux nouveaux médias, les réseaux locaux s'interconnectent de plus en plus.

L'Intranet

Cela commence par les systèmes de messagerie « store and forward » des années 60 et 70 (à distinguer du « store and retrieve ») et dans les conférences sur mainframe et mini, particulièrement dans les universités et la recherche.

La mise en réseau des PC et la normalisation des protocoles répandent la communication dans l'entreprise et entre entreprises. cc:Mail de Lotus et MHS (« Message Handling Service ») de Novell aident à disséminer la messagerie alors que le groupware, avec Lotus Notes, fournit des outils de programmation personnalisée et le lien avec les applications externes.

Les années 90 et 2000 sont celles du temps réel : le groupware a intégré les communications synchrones (conversation, visioconférence, partage d'applications). Ces fonctionnalités seront finalement absorbées par l'Intranet qui s'installe dans les entreprises dans la foulée du Web.

Le Macintosh

En lançant le Macintosh en 1984, Apple a transformé pour toujours les ordinateurs en introduisant trois avancées technologiques (déjà utilisées dans la ligne de produits Lisa) : l'interface graphique ; la souris ; les disquettes 3.5 pouces à haute capacité et très solides.

La conception du Lisa et du Mac doit beaucoup aux idées du centre de recherche de Xerox à Palo Alto, mais Apple les a retravaillées pour les rendre utilisables par un ordinateur produit en masse.

En refusant d'adapter son système d'exploitation MacOS à d'autres processeurs que la famille 68000 ou d'en vendre la licence à d'autres entreprises, Apple a contraint les fabricants de clones à chercher un autre système d'exploitation. Microsoft s'est engouffré dans la brèche et, comme Apple perdait du temps en explorant trop de sentiers différents, Windows a fini par devenir le système d'exploitation préféré des entreprises et des utilisateurs finals.

Toutefois, la migration vers la puissante puce sur RISC d'IBM pour le Power-PC puis la sortie de la ligne de produits iMac ont maintenu Apple à flot.

Le logiciel ouvert

Frustrée par les logiciels en boîtes chers qui ne tenaient pas les promesses annoncées, la communauté des informaticiens commença dans le milieu des années 90 à expérimenter le modèle du « logiciel ouvert » qui permet à l'utilisateur d'accéder aux parties intimes du système.

Microsoft essayait de définir sa stratégie sur le Web et Solaris était trop cher : les entreprises commencèrent à utiliser des serveurs Web Apache sous Linux.

La RAD

Au début, la RAD (« Rapid Application Development ») n'a fait que resserrer le processus d'« édition - compilation - débogage » qui fut longtemps l'essentiel de la programmation. Dans le milieu des années 80, des produits comme Turbo Pascal de Borland ont accéléré le développement d'applications pour PC sous DOS.

Visual Basic de Microsoft, lancé en 1991, a défini le standard auquel tout nouvel outil doit se conformer (au moins pour la commodité du développeur, si ce n'est en performance ou robustesse des applications). Avec la diffusion de composants logiciels diversifiés, Visual Basic a accéléré une transition que des langages plus élégants avaient seulement promis, ouvrant la voie aux classes Java réutilisables.

La technologie RISC

La technologie RISC (« Reduced Instruction Set Computer ») introduite par IBM sur les PC RT en 1986 a permis aux puces d'atteindre les sommets de la performance informatique. Elle était conçue pour faire plus vite les opérations habituelles et faciliter l'utilisation du microprocesseur. Mais ceux qui proposaient le RISC ont sous-estimé la progression de l'architecture X.86 d'Intel, ainsi que sa base installée en logiciels, outils et compétences.

Alpha de DEC, SPARC de Sun et PowerPC de Motorola ont tenté de prendre l'avantage sur Intel au plan de la vitesse de traitement, puis Intel introduisit les techniques RISC dans le cœur de sa ligne X86. Les puces Pentium, AMD et autres appliquent les principes RISC à l'optimisation interne tout en restant compatibles avec les versions anciennes. Les utilisateurs y ont gagné.

VGA

On avait déjà fait de la couleur sur PC, mais l'inclusion du VGA (« Video Graphics Array ») dans la ligne PS/2 par IBM en 1987 fut un événement important. Le PC passait de 16 à 256 couleurs avec une résolution de 320 pixels par 200. On pouvait aussi faire du 16 couleurs avec une résolution de 640 pixels par 480. Cela favorisa le lancement des GUI (« Graphical User Interface »), de l'édition et des jeux d'arcade sur PC. Même aujourd'hui, VGA est la base de tout adaptateur vidéo sur le marché.

Le Web

Quand Tim Berners-Lee appliqua pour la première fois l'hypertexte à l'Internet et forgea le terme « World Wide Web » en 1990, il cherchait à créer une collaboration facile dans les projets. En fait il a transformé l'Internet universitaire en un média de masse.

Dans les neuf années qui suivirent le premier browser et le premier serveur au CERN le Web a acquis l'ubiquité. L'information et la publication furent ses premiers points forts, renforcés à la fin de 1993 par Mosaic et son utilisation graphique.

Enfin le milieu des affaires perçoit vers 1996 le potentiel du Web et cherche à en tirer parti. L'accroissement de la sécurité a réduit les réticences des clients envers l'achat sur le réseau, tandis que des techniques comme XML facilitent l'utilisation du Web pour les affaires.

Windows

Windows est un exemple de la méthode pragmatique qui a permis le succès de Microsoft. D'abord pale imitation de l'interface graphique du Macintosh, Windows était plein de bogues et handicapé par le DOS sous-jacent. Toutefois Microsoft a montré que la compatibilité ascendante, l'attention à des besoins peu intellectuels mais tirés par la productivité, le support aux développeurs et l'OEM (« Original Equipment Manufacturer ») sont plus importants pour le succès d'un système d'exploitation que ses qualités purement techniques.

Il en résulta l'adoption massive de Windows sur les PC. Microsoft appliqua en 1993 la même formule pour faire adopter Windows NT par les serveurs et stations de travail et il a mis Novell et Unix sur la défensive.

3.5.1 Quelques réussites exemplaires

Federal Express

En 1984, FedEx a lancé « Supertracker », fondé sur le « tracking » des paquets qu'il transporte. Ce système permettait aux coursiers sur le terrain d'utiliser des outils communicants manuels et un réseau radio pour saisir le statut et la position des paquets. Les clients peuvent connaître par appel téléphonique la localisation de leurs paquets et la date de livraison attendue.

Le Groupware

En 1989, les responsables du système d'information de Price Waterhouse ont fait un choix audacieux : ils ont supprimé la messagerie électronique qu'ils venaient d'installer sur 10 000 postes et l'ont remplacée par Lotus Notes, produit nouveau que personne ne comprenait et qu'ils déployaient sous OS/2. Price Waterhouse a ainsi été la première grande entreprise à déployer Lotus Notes au niveau mondial. Cela lui a permis de gérer ses projets globalement.

Le groupware a décollé, notamment dans les entreprises de service. Coopers & Lybrand - qui a fusionné avec Price Waterhouse en 1998 - avait installé Lotus Notes en 1993, ce qui a facilité la fusion des deux entreprises.

L'interface graphique

En 1990, au milieu d'une grave crise du transport aérien, United Airlines a jugé rentable de remplacer ses terminaux « bêtes » par des PC dotés d'une interface graphique. Jugeant Windows trop instable, United a développé sa propre interface graphique. Le système, installé sur des PC 80206, a permis de réduire le délai de traitement d'une réservation de 10 %, la durée de la formation de 25 %, et d'économiser ainsi 9 millions de dollars dès la première année.

Le Datawarehouse

Dans le milieu des années 80 Wal-Mart voulait s'étendre au delà du Middle West. Mais comment gérer plusieurs magasins en tenant compte des particularités de chaque marché local ? Les managers de Wal-Mart eurent l'idée de collecter les données sur les ventes et de les utiliser pour répondre rapidement à des changements de tendance de la demande. Il en résulta un des plus grands succès du datawarehouse. Le système grimpa rapidement à 700 Go. Il permit au détaillant de partager l'information sur la demande avec ses fournisseurs et de leur faire gérer ses stocks. Wal-Mart surpassa ainsi Sears et Roebuck. Son datawarehouse, exploité sur du matériel NCR Teradata, a crû jusqu'à 24 To (téraoctets, ou milliers de Go).

Commerce électronique

En 1995, Barnes & Noble et Crown Books développaient des réseaux de librairies concurrents. Jeffrey Bezos, informaticien chez Bankers Trust à Chicago, se lança sur l'Internet pour créer Amazon.com qui sera la première réussite du commerce électronique.

Amazon a montré que l'on peut faire du commerce électronique malgré les inquiétudes concernant la sécurité sur le réseau. Elle offre plusieurs millions de titres de livres en ligne, des CD, des livres audio, des DVD et des jeux pour ordinateur.

Par ailleurs Dell a prouvé qu'il était rentable d'intégrer le commerce électronique avec le traitement d'une chaîne d'approvisionnement pour la distribution des micro-ordinateurs. Lancé au début de 1997, le site de Dell donne aux clients des dates de livraison exactes en contrôlant les stocks de pièces détachées lors de la commande. Dell partage les informations sur la demande avec ses fournisseurs en temps réel, ce qui permet de réduire les coûts d'immobilisation et d'améliorer le service au client.

3.6 Les langages de programmation

Tout programme est écrit dans un langage qui, traduit par un compilateur ou interprété par un interpréteur, pourra ensuite être exécuté automatiquement par l'ordinateur. Dès 1967 on dénombrait 120 langages dont seuls 15 étaient vraiment utilisés (Sammet [73], p. vi). Les programmeurs (que l'on appelle aussi « développeurs », le mot programmeur ayant de façon regrettable été jugé péjoratif) utilisent encore aujourd'hui des langages créés dans les années 50, période d'intense innovation.

Certains langages ont été conçus pour le calcul scientifique, d'autres pour la ges-

tion des entreprises, d'autres enfin pour la formalisation du raisonnement ou le calcul algébrique. Il existe aussi des langages étroitement adaptés à une finalité technique précise.

* *

Au tout début, dans les années 40, les programmeurs devaient écrire dans le langage machine de l'ordinateur. Son vocabulaire est constitué de nombres binaires représentant les adresses des mémoires et les codes des opérations. Mais programmer dans un tel langage est très pénible car il offre peu de repères à la mémoire de l'être humain.

L'assembleur, conçu en 1950, permet de coder les opérations en utilisant des caractères alphabétiques (ADD pour l'addition, SUB pour la soustraction etc.) et il traduit ces codes en langage machine. Néanmoins il était nécessaire de définir des langages encore plus commodes, dits « de haut niveau » (ils sont relativement faciles à apprendre et à utiliser, mais leur mise en œuvre sur la machine suppose une cascade de traductions).

Le premier langage « de haut niveau » fut Fortran (« Formula Translation ») conçu par John Backus à IBM en 1954. Ses instructions ressemblent à des formules de mathématiques et il est bien adapté aux besoins des scientifiques, mais incommode pour les travaux peu mathématiques et notamment pour programmer des logiciels de gestion. IBM considérait Fortran comme un langage « propriétaire » qui devait être utilisé uniquement sur ses machines. Algol a été développé en 1958 par un consortium européen pour le concurrencer.

Le Cobol (« Common Business Oriented Language », développé en 1959 par un consortium comprenant le Department of Defense) était destiné aux logiciels de gestion. Délibérément « verbeux », le Cobol emploie des mots et une syntaxe proches de l'anglais courant.

D'autres langages encore plus commodes furent introduits ensuite : Basic (« Beginner's All-Purpose Symbolic Instruction Code », 1964) peut être rapidement maîtrisé par le profane ; il est utilisé dans les écoles, entreprises et ménages. C (1972) est un langage de haut niveau, mais il peut aussi être utilisé comme un assembleur car il permet de programmer des instructions au plus près de la « physique » de la machine. Beaucoup de logiciels pour les entreprises seront écrits dans ce langage souple dont l'utilisation est dangereuse pour le débutant : comme il permet de tout faire, il comporte peu de « garde-fous ».

Pascal (1970), langage « structuré » conçu de façon à éviter les erreurs de programmation en encourageant la modularité, sera largement utilisé par les pédagogues qui veulent donner aux étudiants une première formation à la programmation.

Certains langages de haut niveau sont adaptés à des applications précises : APT (« Automatically Programmed Tools ») pour le contrôle des machines outils numériques, GPSS (« General Purpose Simulation System ») pour la construction des modèles de simulation, LISP¹² pour manipuler des symboles et des listes (suites de symboles) plutôt que des données. LISP sera souvent utilisé en intelligence artificielle. Scheme est, parmi les dialectes de LISP, celui qui rassemble le plus de partisans.

12. « List Processing », créé par John MacCarthy au MIT en 1959.

Perl¹³ est un langage de commande commode dans le monde Unix et pour les serveurs Web. Les langages de quatrième génération (4GL), utilisés surtout pour la gestion et l'interrogation des bases de données, seront encore plus proches du langage humain. On peut citer Focus, SQL (« Structured Query Language ») et dBase.

Les langages à objets (que l'on appelle aussi « langage orientés objets », expression inutilement lourde) comme Simula (1969), Smalltalk (1980)¹⁴, C++ (1983)¹⁵ ou Java (1995)¹⁶ permettent d'écrire des logiciels fondés sur des « objets », ce mot désignant de petits programmes rassemblant quelques données et traitements et qui communiquent entre eux par des messages (voir page ??). La logique des langages objet est proche de celle de la simulation. L'évolution des langages objet est allée vers la simplicité et la sécurité : Java contient plus de « garde-fous » que C++, qui comporte les mêmes risques que C.

Les programmes peuvent s'articuler autour d'un « bus » qui assure notamment le routage des messages et soulage d'autant l'écriture de chaque programme : on parle alors d'EAI (« Enterprise Application Integration ») ou encore de « middleware » (voir page ??).

Le système d'exploitation (ou OS pour « Operating System ») est un programme qui assure la mobilisation des ressources physiques de la machine : allocation du processeur et de la mémoire vive aux divers programmes en cours d'utilisation, gestion automatique des entrées et sorties vers les périphériques (disque dur, imprimante, modems, « ports » de communication), messages d'erreurs, exécution des tâches de fond etc. La première version d'Unix a été développée en 1969 (Raymond [70]), les bases de données relationnelles sont nées en 1970. Linux est une version d'Unix adaptée au PC ; il concurrence fortement Windows, en particulier sur le marché des serveurs.

Le mot « programmation » recouvre des activités diverses : l'utilisateur individuel « programme », même s'il ne s'en rend pas compte, lorsqu'il utilise Excel et Word ; il peut aussi, s'il a un tempérament de bricoleur, faire de petits programmes en Pascal ou en Scheme : mais dans la plupart des cas ce bricolage n'ira pas loin même s'il est ingénieux. Les gros programmes sont écrits par des équipes de programmeurs spécialisés qui se partagent les tâches et utilisent souvent des générateurs de code (comme Rational Rose) pour la partie la plus mécanique du travail d'écriture. La différence entre le programme individuel et le gros programme est du même ordre que celle qui existe entre le travail (éventuellement réussi) qu'un bricoleur bien équipé peut réaliser à domicile et la construction d'une automobile ou d'un avion, qui suppose une maîtrise de diverses techniques dont seule une très grande entreprise peut être capable.

L'art de la programmation fait l'objet d'une abondante littérature, qu'il s'agisse d'indiquer des règles de bon sens mais qui ne sont faciles ni à découvrir, ni à mettre en pratique (Arsac [3]), d'introduire à l'informatique à travers la maîtrise d'un langage (Abelson et Sussman [26], Bloch [5], Felleisen et alii [58]), d'explorer à fond le monde des algorithmes (Knuth [51]) etc.

13. Créé par Larry Wall en 1987.

14. Créé par Alan Kay (Hiltzik [43]) au PARC de Xerox.

15. Créé par Bjarne Stroustrup (Stroustrup [79]) aux Bell Labs.

16. Créé par Scott McNealy (Stiller et Marlowe [30]) chez Sun.

3.7 Le marché de l'informatique

La vivacité de la concurrence entre les fournisseurs contribue à expliquer la baisse du prix des ordinateurs. Les progrès des systèmes d'exploitation, applications, processeurs et mémoires sont autant d'occasions pour relancer la concurrence, différencier les produits (tout en restant compatible avec le standard PC) et se tailler de nouvelles niches de marché.

En 1949, une étude de marché d'IBM estimait que neuf SSEC pourraient satisfaire les besoins des États-Unis pendant plusieurs décennies (le SSEC, construit en 1948 et rival de l'ENIAC, fut l'un des premiers ordinateurs à respecter l'architecture de von Neumann (*Perspectives of the Founding Fathers* [1]) ; on dénombrait en 2000 plus de 500 millions d'ordinateurs dans le monde et depuis 1995 il se vend chaque année plus de PC que de téléviseurs.

L'amélioration des techniques alimente une baisse du prix (voir page ??) qui induit un élargissement du marché ; cet élargissement provoque une nouvelle baisse de prix etc. L'offre se diversifie du « desktop » au « laptop » portable, puis au « palmtop » qui tient dans le creux de la main et qui, muni d'une antenne, apporte toutes les facilités du PC en réseau.

Le micro-ordinateur a transformé dans les années 80 le marché de l'informatique (Grove [41]). Au début de cette décennie, des entreprises comme IBM, DEC, Wang ou Univac étaient organisées verticalement, chacune étant présente dans toutes les couches (puces, ordinateurs, systèmes d'exploitation, applications, vente et distribution). L'industrie informatique était « propriétaire », un système entier étant produit par un seul fournisseur (figure 3.3).

L'avantage de cette organisation du marché, c'est que l'offre d'un fournisseur constitue un ensemble cohérent. L'inconvénient, c'est qu'une fois le fournisseur choisi le client est contraint de lui rester fidèle car s'il veut prendre un autre fournisseur il devra changer tout son système informatique d'un coup, ce qui est pratiquement impossible. La compétition pour la première vente à un client est donc féroce.

En 1995, la structure du marché n'est plus la même. La baisse des prix a fait entrer l'informatique dans l'ère de la production de masse. Le client peut faire intégrer des éléments (puce, système d'exploitation, applications) fournis par des entreprises différentes. De nouvelles compagnies se sont créées (Intel, Microsoft, Compaq etc.), spécialisées chacune dans une couche où elle se trouve en concurrence avec d'autres entreprises spécialisées (figure 3.4).

Les stratégies diffèrent selon le secteur d'activité. Les fournisseurs de matériel font l'essentiel de leur profit sur leurs produits les plus récents ; ils souhaitent que l'innovation rende le matériel rapidement obsolète afin que les clients le renouvellent fréquemment. Ils voudraient que les fournisseurs de logiciels utilisent à fond les dernières possibilités du matériel, mais seuls quelques millions d'ordinateurs en sont équipés. Pour pouvoir toucher un large marché, les fournisseurs de logiciels ont intérêt à produire des programmes qui pourront tourner convenablement sur les machines anciennes : ils sont donc plus « conservateurs » que les fournisseurs de matériel (Colwell [17] p. 33).

Ainsi pendant les années 80, et sans que l'on puisse assigner de date précise au changement, le marché s'est restructuré. IBM, entreprise phare du début des années 80, a subi une crise profonde ; les premières places ont été prises par les Microsoft,

Intel, Compaq, Dell, Novell etc.

3.7.1 Dynamique des équilibres du logiciel

Pour comprendre la succession des équilibres sur le marché des logiciels, il faut remonter jusqu'aux années 60 puis situer les dates du passage d'un équilibre à l'autre (le repérage de telles dates charnières est toujours délicat : nous identifierons les événements caractéristiques sans attacher trop d'importance à la précision des dates).

Dans les années 60, la pratique du logiciel libre, à la fois gratuit et ouvert (le code source est mis gratuitement à disposition) est répandue sans être théorisée. Elle est mise en œuvre par des « hackers » (voir page 85).

On peut dater de 1969 l'émergence du marché du logiciel. Jusqu'à cette date, IBM vendait conjointement le matériel *avec* le logiciel (systèmes d'exploitation, langages de programmation, applications) qui permettait de faire fonctionner le matériel. En 1969, pour éviter que les concurrents ne bénéficient gratuitement des logiciels qu'il avait produits - et aussi pour se prémunir contre des procès pour cause de monopole - IBM a mis en place la politique d'« unbundling » qui introduisait une tarification séparée du matériel et du logiciel.

Par ailleurs à partir de la publication de la lettre ouverte de Bill Gates en 1976 (voir page 89), le commerce du logiciel pour PC compilé et vendu comme une boîte noire se met en place.

On peut dater de 1991 - date de la première version de Linux, mais aussi date de naissance du World Wide Web - la résurgence du modèle du logiciel ouvert (voir Open Source Initiative, « Open Source Definition¹⁷ », 1999). Il se développera en s'appuyant sur l'Internet et en tirant argument des inconvénients que présente l'économie du logiciel compilé.

Cependant entre 1976 et 1991, le logiciel ouvert n'avait pas disparu ; il a constitué un thème de militantisme pour des personnes comme Richard Stallmann (Stallman [78]) ou Eric Raymond (Raymond [69]) ; mais s'il avait un rayonnement intellectuel sa viabilité économique semblait douteuse. Il n'en est plus de même maintenant : le modèle du logiciel ouvert a gagné en crédibilité durant les années 90, il a conquis des parts de marché¹⁸ et son avenir est garanti. Il reste à situer la ligne selon laquelle le marché se partagera entre les deux modèles.

L' « Unbundling » d'IBM

Le « consent decree » de 1956 entre IBM et le département de la justice oblige IBM à offrir à la vente les matériels qu'il louait auparavant, à facturer séparément les pièces détachées et la maintenance, et à publier l'information sur la maintenance de ses machines de telle sorte qu'elle puisse être exécutée par d'autres entreprises.

Des entreprises se créent alors : elles achètent des machines IBM et les vendent en leasing à un prix inférieur au loyer que pratique IBM. Pour les contrer IBM accélérera

17. www.opensource.org/osd.html

18. Notamment le marché des serveurs Web : en 2000, 60 % des serveurs utilisaient Apache contre 20 % à Microsoft IIS et 7 % à Netscape Enterprise. Le nombre des utilisateurs de Linux dans le monde se situerait à la même date dans la fourchette de 7 à 16 millions.

le rythme d'introduction des nouvelles machines. Par ailleurs la publication des informations sur la maintenance avait rendu visibles les interfaces entre machines IBM. RCA, Amdahl et Fujitsu ont alors pu offrir des ordinateurs compatibles avec les machines IBM et de nombreuses entreprises ont offert des périphériques compatibles (écrans, lecteurs de disques, lecteurs de bandes et imprimantes).

IBM offrait par ailleurs à ses clients ses programmes et services d'ingénierie sans les facturer séparément : leur coût était inclus dans le prix du matériel, l'ensemble constituant un « paquet » (« bundle »). Cette formule simplifiait la vie des clients mais les acheteurs de matériels compatibles disposaient gratuitement de ces services dont les concurrents d'IBM n'avaient pas supporté le coût.

Dès 1964, le Spectra 70 de RCA a été perçu comme une menace pour l'IBM 360 ; mais « les ingénieurs de RCA n'ont pas vu l'importance de la compatibilité et n'ont pas su résister à la tentation d'introduire des “améliorations” par rapport au S/360. Il en résulta que les machines de RCA ne pouvaient pas utiliser sans modification les programmes en S/360. Cette conversion étant difficile, peu de clients achetèrent les systèmes de RCA » (Humphrey [45] p. 59.)

Il semblait impossible de protéger les logiciels par des brevets ; les experts d'IBM conçurent donc en 1966 un système associant copyright et licence d'utilisation.

En janvier 1969, le département de la justice engagea des poursuites contre IBM en application de la loi antitrust. L'« unbundling » fut annoncé par IBM en juin 1969 : moyennant une réduction de 3 % du prix du matériel, les logiciels seraient vendus désormais à part. Par la suite, lorsque des entreprises attaquèrent IBM en justice en s'appuyant sur les dispositions antitrust, IBM gagna tous les procès en tirant argument de l'unbundling.

L'unbundling a transformé le commerce du logiciel et de l'ingénierie. IBM Global Services est devenue la partie la plus importante et la plus profitable d'IBM. L'interface de programmation (API) s'est stabilisée pour répondre aux besoins de compatibilité des utilisateurs, et cette stabilité a favorisé la diversification de l'offre.

Naissance du marché du logiciel compilé

Le marché du logiciel compilé pour PC, utilisable mais illisible pour l'utilisateur, démarre avec la « Open Letter to Hobbyists » publiée par Bill Gates le 3 février 1976 (voir page 89).

Bill Gates avait alors 20 ans. Ce jeune homme avait un fort potentiel, comme on l'a vu par la suite : d'une part sa compétence en informatique faisait de lui un bon hacker ; d'autre part - et contrairement à la majorité des autres hackers - son milieu social l'avait initié aux affaires. Son père était l'un des plus grands avocats d'affaires de Seattle ; sa mère siégeait au conseil d'administration de plusieurs grandes entreprises et universités. Il était ainsi mieux préparé que d'autres à percevoir le potentiel marketing de la vente du logiciel compilé, et peut-être aussi le fait que seule une telle organisation du marché pourrait fournir la diversité des logiciels dont les micro-ordinateurs allaient avoir besoin.

Les hackers sont restés désarmés devant son attaque. Gates les avait pris en tenaille entre deux cultures américaines : celle du pionnier qui va de l'avant dans des territoires vierges et s'y sert de l'« open source » pour se débrouiller ; celle de la libre entreprise, qui ne peut pas se concevoir sans une protection du droit de propriété.

C'était le renouvellement de la lutte entre le fermier et les cow-boys !

Gates a défini le modèle économique qui s'imposera sur le marché des logiciels pour PC. Il a créé une industrie du logiciel dont il est devenu le plus grand dirigeant. Le modèle qu'il a inventé était sans doute alors le seul possible et sa fécondité suscite l'admiration, quelle que soit l'opinion que l'on peut avoir sur la qualité des produits de Microsoft.

Limites du logiciel compilé

Cependant ce modèle a une limite qui se révèle avec la complexification des logiciels. Elle est due notamment aux exigences de la « compatibilité ascendante » : la version V_n d'un logiciel doit en principe être capable de lire et de traiter les fichiers composés avec les versions V_{n-k} antérieures. En passant d'une version à l'autre, le logiciel s'alourdit ainsi de conventions anciennes dont il doit garder la trace.

La complexification est due aussi à la cible marketing elle-même : pour que le logiciel puisse couvrir un large marché, il doit offrir une grande diversité de fonctionnalités et pouvoir être exécuté sur des plates-formes diverses. Chaque client n'utilisera donc qu'une toute petite partie du logiciel qu'il a acheté et dont le volume encombre la mémoire vive de son ordinateur.

Or plus un logiciel est compliqué, plus il est difficile à « déboguer ». Le logiciel est une construction fragile. L'éradication des bogues est d'autant plus difficile que l'on risque toujours, en corrigeant une erreur, d'en introduire d'autres. C'est une tâche qu'il est impossible d'automatiser complètement (voir page ??).

On ne peut physiquement plus désormais, même avec 30 000 programmeurs groupés sur un campus (plus quelques sous-traitants qui coopèrent à travers l'Internet) faire converger convenablement le processus de « débogage » d'un grand logiciel comme Windows. Les versions successives sont commercialisées avec leurs bogues et les utilisateurs subissent des incidents désagréables.

Organisation de la production d'un logiciel complexe

Pour réaliser une construction intellectuelle complexe comme un système d'exploitation ou un grand logiciel applicatif, il faut réunir les quatre éléments suivants :

- un *centre* capable d'attirer les contributeurs et de sélectionner les contributions à retenir ;
- des *contributeurs* qui réalisent le travail de développement ;
- un *réseau* de communication ;
- un moyen de *rémunérer* les contributeurs.

Le modèle « Microsoft » a permis de rassembler la masse critique de contributeurs nécessaire à la production des logiciels pour PC. Le centre, c'est Bill Gates lui-même, assisté par une équipe de proches collaborateurs ; le réseau, c'est le réseau local du campus de Redmond et l'Internet entre Microsoft et ses sous-traitants ; les contributeurs sont les programmeurs de Microsoft et les entreprises sous contrat ; la rémunération est financière.

Dans le modèle « Linux », le centre est Linus Torvalds assisté par une équipe de proches collaborateurs ; le réseau est l'Internet, les contributeurs sont potentiellement tous les développeurs du monde entier ; la rémunération est symbolique (ce qui

ne veut pas dire qu'elle soit irréaliste), ce qui implique que le centre soit un « dictateur bienveillant » (Tirole et Lerner [31]).

Le modèle « Linux »

La culture finlandaise de Linus Torvalds¹⁹ le libère des dogmes de la libre entreprise à l'américaine. Il tire parti de l'Internet pour utiliser le mode de développement « open source » : le code source de Linux est disponible sur l'Internet, ainsi d'ailleurs que ceux de Java, Perl etc.

Quiconque repère une bogue peut, s'il en a la compétence, proposer une correction qui sera soumise au « dictateur bienveillant ». L'Internet élargit à des millions de programmeurs le cercle des contributeurs potentiels ; il permet d'accélérer la convergence du débogage et de poursuivre sans fin le processus de perfectionnement. Il permet aussi à l'utilisateur de choisir « à la carte » les fonctionnalités dont il a besoin, ce qui réduit la taille des logiciels.

Se crée alors, en contraste avec l'économie marchande du logiciel compilé, une économie indirecte de la reconnaissance professionnelle autour du logiciel libre²⁰ : « indirecte » parce que si un développeur contribue gratuitement à Linux il sera respecté dans son entreprise et prendra de la valeur sur le marché. L'économie de la reconnaissance est une économie symbolique, mais non une économie de la gratuité : le symbole est rémunérateur à la fois psychologiquement et financièrement²¹.

D'ailleurs « logiciel libre » ne signifie pas « logiciel gratuit », mais « logiciel dont le code source est lisible ». Une économie marchande s'est bâtie sur le logiciel libre, qui se vend toutefois à des prix inférieurs à ceux du logiciel compilé.

Le retour de la logique « logiciel libre (pas cher) » met en péril la logique « programme compilé (cher) en boîte ». L'issue de la lutte est prévisible : la qualité est du côté du logiciel libre qui seul permet de faire converger le débogage ou du moins accélère sa convergence. On peut compter sur le talent stratégique de Bill Gates : il saura adopter sagement le système de l'adversaire, même si pour le moment Microsoft résiste bec et ongles en tentant de convaincre les acheteurs et les administrations que le logiciel libre est dangereux. Linus Torvalds le dit avec philosophie : « J'aurai gagné quand Microsoft se mettra au logiciel libre ».

Limites du logiciel libre

Le logiciel libre a lui aussi ses limites : il arrive qu'il se diversifie en variantes et qu'il soit déficient du point de vue de la documentation, des interfaces utilisateurs ou de la compatibilité ascendante. Certains disent que le logiciel libre est destiné

19. Les grands créateurs de logiciels ont chacun des racines culturelles profondes : les écrits de Bjarne Stroustrup (C++), danois, rappellent l'existentialisme protestant de Kierkegaard ; Linus Torvalds (Linux), finlandais, a été marqué par l'épopée du Kalevala ; Niklaus Wirth (Pascal), suisse, par le calvinisme ; Larry Wall (Perl) et Alan Kay (Smalltalk), tous deux américains, ont cherché leurs références culturelles dans le monde de l'enfance. Voir « Interview with Linus Torvalds: What Motivates Free Software Developers? » www.firstmonday.dk/issues/issue3_3/torvalds/index.html, 1999

20. « The « utility function » Linux hackers are maximizing is not classically economic, but is the intangible of their own ego satisfaction and reputation among other hackers. » (Raymond [69]).

21. Les mécanismes de reconnaissance sont ici semblables à ceux de la recherche universitaire.

Le « dictateur bienveillant »

Dans le monde du logiciel libre, la production d'un gros programme ne peut aboutir que si elle est animée par un « dictateur bienveillant ». Le talent nécessaire pour jouer ce rôle est donc crucial.

L'expression « dictateur bienveillant » est bien sûr paradoxale. Il faut la comprendre ainsi : beaucoup de personnes (à vrai dire, tout le monde) peut suggérer des modifications du code, mais seule une petite équipe est autorisée à introduire effectivement des modifications : c'est là le côté « dictateur ».

Cependant cette petite équipe manifeste de la considération aux personnes qui font des suggestions, même et surtout si la suggestion n'est pas retenue : c'est cela le côté « bienveillant ».

Si le « dictateur » était désagréable ou méprisant, il ne recevrait plus de suggestions et le processus d'amélioration serait stoppé.

à des experts, alors que Microsoft a eu l'intelligence de faire des produits destinés à l'utilisateur non expert²² : en effet les administrateurs des serveurs Apache ou Sendmail sont plus experts que l'utilisateur moyen d'un PC.

Les statistiques montrent que les contributeurs vraiment actifs sont peu nombreux et que la plupart des contributions sont très simples (ce qui ne veut pas dire qu'elles ne soient pas précieuses : l'alarme qui indique une bogue est toujours bienvenue). Pour Apache, 83 à 91 % des modifications proviendraient de 15 programmeurs (Mockus, Fielding et Herbsleb [29]).

Le logiciel libre et les pirates

Certains disent que nous allons vers un « Pearl Harbor électronique » (Linda Dailey Paulson [65] p. 22). Pourquoi ? parce que les pirates (terme qu'il faut préférer à « hackers » pour ne pas les confondre avec les pionniers des années 60 et 70) progressent, alors que la recherche sur la sécurité piétine.

Les pirates progressent *parce qu'ils utilisent les méthodes du logiciel libre* : ils diffusent le code source de leurs programmes et en partagent les améliorations. La virulence de leurs produits augmente, leur compétence s'accroît, le nombre et l'ingéniosité des attaques augmentent. Leur activité devient paraît-il rentable : les pirates racketteraient des entreprises vulnérables à qui ils vendent leur « protection », utilisant ainsi les recettes éprouvées de la mafia.

Pour faire face aux pirates, on compte seulement 200 chercheurs compétents en sécurité dans les universités et entreprises américaines. Dès qu'un chercheur devient compétent, il est embauché par une entreprise qui veut utiliser son savoir-faire pour se protéger : il en résulte une pénurie de professeurs qui explique le faible nombre de

22. « In every release cycle Microsoft always listens to its most ignorant customers. Linux and OS/2 developers, on the other hand, tend to listen to their smartest customers... The good that Microsoft does in bringing computers to non-users is outdone by the curse that they bring on experienced users. » (Tom Nadeau, « Learning from Linux » www.os2hq.com/archives/linmemo1.htm, 1999).

diplômés experts en sécurité. De façon paradoxale, la vigueur de la demande assèche ici la formation des compétences.

Ainsi la lutte est inégale : d'un côté les pirates tirent parti des méthodes de production du logiciel libre, construisent leur rentabilité et se multiplient ; de l'autre, la recherche est entravée par le copyright, vidée de ses compétences par l'appel du marché, et elle ne suit pas une politique d'ensemble. Une simple extrapolation de cette tendance désigne le futur vainqueur : c'est très inquiétant.

On observe toutefois chez les pirates une certaine retenue : ils pourraient bloquer tout l'Internet mais ils n'en font rien. La plupart d'entre eux sont des joueurs que leur petit jeu les amuse beaucoup, et ils ne pourraient plus s'amuser s'ils bloquaient le terrain de jeu. Mais il suffirait qu'il se trouvât parmi eux un pervers, quelqu'un qui prenne plus de plaisir à nuire qu'à jouer (ou qui ait, pour des raisons quelconques, un fort intérêt à nuire) pour que le danger devienne réalité.

La tentation de l'« enclosure »

Certaines des innovations permises par l'Internet (moteurs de recherche gratuits, échanges de fichiers audio et bientôt vidéo, création coopérative de bases de données) ont compromis l'équilibre économique des détenteurs de droits et des diffuseurs de contenus. Il en résulte un développement des techniques visant à instaurer un contrôle : « policy based routing » de Cisco ; « réseaux intelligents » des opérateurs télécoms. La vente des copies de logiciels peut être remplacée par un transfert temporaire du droit d'utilisation d'après le projet d'UCITA (Uniform Computer Information Transaction Act) (Syme et Camp [32]). Les logiciels peuvent être protégés par des brevets depuis 1980 : en 1999, les brevets sur les logiciels ont représenté 14 % du nombre des brevets aux États-Unis. Le DMCA (Digital Millenium Copyright Act) de 1998 renforce la protection des données informatiques : toute tentative de décryptage est interdite. La durée du copyright augmente, passant de 14 ans à 70 ans après la mort de l'auteur.

Michel Gensollen [38] propose de redéfinir droits de propriété sur les biens informationnels de telle sorte que leur coût de production puisse être équilibré par des recettes tout en protégeant la facilité d'utilisation : il faudrait ainsi distinguer le droit d'accès, l'usufruit, le droit d'administration, le droit d'exclusion (déterminer qui a accès au bien), le droit d'aliénation. La rémunération devrait reposer sur des redevances plutôt que sur des droits de propriété. De nouveaux acteurs devraient émerger pour assurer l'animation des clubs de producteurs et le partage des droits.

Sur ce marché, l'innovation juridique suit l'innovation technique qui elle-même va s'orienter vers les canaux que le droit protège le mieux. Nous sommes là au tout début d'une dialectique qui ressuscite, sur le réseau, les stratégies qu'ont utilisées jadis Venise, les villes de la Hanse, les navigateurs et commerçants du Portugal et des Pays-Bas etc.

3.8 Les logiciels de bureautique

Nous utilisons chaque jour les logiciels de bureautique. D'où viennent-ils ? Comment ont-ils été conçus ? Quels sont les problèmes qu'ils posent ?

La bureautique a longtemps été considérée avec méfiance par les informaticiens :

qu'il soit possible d'utiliser l'ordinateur sans avoir à le programmer, cela ouvrait une brèche dans leur monopole et, il faut aussi le dire, cela créait un certain désordre.

Mais si l'on aborde le système d'information en adoptant le point de vue de ses utilisateurs on ne peut pas négliger la bureautique. Comprendre d'où elle vient, comment elle a été conçue, permet de mieux évaluer les conditions de sa bonne utilisation et de voir les causes de certaines erreurs.

S'approprier au micro-ordinateur

Les épisodes d'apprentissage sont peu glorieux ; on se hâte de les oublier parce qu'on s'y est trouvé maladroit et un peu ridicule. Ils occupent pourtant une place importante dans les soucis des utilisateurs.

Je me rappelle avoir reçu mon premier PC (un IBM PC AT) en octobre 1987 alors que je travaillais au CNET (Centre National d'Études des Télécommunications) à Issy-les-Moulineaux. Auparavant j'utilisais un terminal Scorpion relié au calculateur du centre de calcul pour écrire (en Fortran) et faire tourner des programmes de calcul économique. Ces programmes, je l'avoue, étaient riches en boucles et en instructions GOTO : bâtis autour d'une idée initiale simple, ils s'étaient transformés en plats de spaghettis et j'avais quelque peine à les maîtriser lorsque je voulais programmer des variantes.

Les micro-ordinateurs étaient déjà répandus mais je n'en avais pas éprouvé le besoin (je ne suis pas de ceux qui se jettent sur la nouveauté) jusqu'au jour où Alain Le Diberder, du BIPE, m'a montré comment il utilisait Multiplan sur son PC : j'ai été intrigué, puis séduit par la puissance et la simplicité de l'outil.

Avoir sur son bureau non plus un terminal, mais un ordinateur complet avec mémoire et processeur alors qu'auparavant je n'avais jamais pu toucher la machine, c'était sensationnel mais intimidant. En particulier la disquette, qui permettait d'introduire de nouveaux programmes dans mon PC ou de transférer des fichiers d'un PC à l'autre, m'impressionnait beaucoup : j'hésitais à l'utiliser car pénétrer dans l'ordinateur me semblait proche du sacrilège, fait que je livre à la sagacité des psychanalystes amateurs.

Je trouvais les commandes de MS-DOS compliquées et peu naturelles. Comment faire marcher, d'ailleurs, une machine que je ne savais pas programmer ? Gérard Dubois m'éclaira quand il me dit « sur le PC, on utilise trois logiciels : le tableur, le traitement de texte et le logiciel graphique » (à l'époque, il n'y avait pas au CNET de réseau local, donc ni messagerie ni documentation électronique). Dubois avait raison, puisque c'est bien ainsi que la plupart des utilisateurs se servent de leur PC ; mais cette phrase éteignit en moi l'ambition de programmer le PC et me détournait pour longtemps de l'invite de commandes (cette fenêtre où figure le symbole `C:\>` que l'on appelle le « prompt »). J'aurai par la suite l'occasion de le regretter.

J'ai muni sur ses conseils mon PC du traitement de texte Textor, du tableur Multiplan et d'un logiciel graphique dont j'ai oublié le nom car je ne l'ai pratiquement jamais utilisé. Pour m'approprier au traitement de texte et au tableur j'ai dû passer des heures à lire les notices et à m'imprégner de conventions étranges (feuilles de style, macros, liens etc.) Regarder par dessus l'épaule des collègues était utile pour découvrir de nouvelles astuces (pour effacer un caractère, faut-il utiliser le retour arrière ou la touche « Suppr » ? Comment nommer les feuilles d'un tableur ? etc.)

Programmer en Multiplan m'a conduit à abandonner la programmation en Fortran sans pour autant devenir un virtuose du tableur (j'avais conservé le terminal pour pouvoir utiliser mes anciens programmes et j'ai continué à utiliser le Fortran pour les variantes). Le traitement de texte me permit de soulager Claude Bernard qui jusqu'alors avait tapé mes manuscrits. En m'entraînant je suis peu à peu devenu un dactylographe convenable : c'est sans doute ce qu'il y a eu de plus réussi dans mon apprentissage²³.

En janvier 1989 j'ai quitté le CNET pour créer avec d'autres ingénieurs Arcome, entreprise de conseil en télécoms. Antoine Laurès était un spécialiste des réseaux locaux. Fervent du Macintosh, il nous a convertis et nous avons pu découvrir ainsi une ergonomie proche de celle que l'on trouve aujourd'hui sur Windows : souris, menus déroulants, icônes, poubelle etc.

Arcome a relié ses Macintosh par un réseau local Appletalk. Nous avons découvert la messagerie, l'agenda partagé, le transfert de fichiers, le partage des imprimantes. C'était une expérience étonnante ! Je ne sais pourquoi, il m'a été plus facile d'admettre ces nouveautés-là que la disquette ; travailler à plusieurs sur un plateau paysagé m'y a peut-être aidé car l'échange d'astuces était intense. Cette modeste réalisation nous ayant donné de l'avance par rapport à nos clients, nous avons pu les conseiller en connaissance de cause.

En février 1990 j'ai quitté Arcome²⁴ pour créer Eutelis, autre société de conseil²⁵. J'y ai installé un réseau Ethernet pour relier les Macintosh. Un an après, l'entreprise s'est mise au PC pour utiliser la même machine que ses clients. Philippe Penny, venu du CNET comme directeur à Eutelis, nous a apporté son expertise en bureautique communicante (ou « groupware ») et nous a fait découvrir la gestion électronique de documents sous Lotus Notes (Henry [42] ; Penny et Volle [33]). En utilisant la réplication entre des serveurs locaux, Lotus Notes permettait de partager une même documentation entre plusieurs sites, ce qui anticipait sur l'Internet. Nous avons ainsi pu expérimenter l'Intranet avant que ce terme n'existe et l'installer chez nos clients.

En 1995 nous nous sommes intéressés à l'Internet auquel cette expérience nous avait préparés. Eutelis a utilisé la messagerie sur l'Internet et créé un site Web. Nous avons étudié l'économie de l'Internet et découvert les perspectives du commerce électronique. En août 1998, m'appuyant sur cette expérience, j'ai créé le site www.volle.com et utilisé FrontPage pour y publier mes travaux.

Parallèlement aux expériences que nous faisions dans notre entreprise nous avons mis en place des systèmes d'information chez nos clients. Pour certains d'entre eux, le groupware fournissait des solutions élégantes, efficaces et peu coûteuses. Il était cependant difficile d'en convaincre les informaticiens, qui se méfiaient de la bureautique et préféraient des projets plus lourds. Par ailleurs les changements qu'introduisaient dans l'organisation du travail les workflows, la rédaction coopérative, la dissémination sélective etc. (Winograd et Flores [25]) n'étaient pas toujours les bienvenus, malgré leur utilité. Enfin les utilisateurs rencontraient naturellement les mêmes dif-

23. La méthode que j'ai utilisée est décrite à la page www.volle.com/travaux/dactylo.htm.

24. J'ai découvert à cette occasion la recette infallible pour se brouiller avec son meilleur ami : s'associer avec lui pour créer une entreprise.

25. Eutelis était filiale d'un grand groupe. J'ai appris à Eutelis que le plus gros des trois mensonges de la vie en entreprise est « je viens de la holding pour vous aider » (les deux autres, plus banals, sont « on vous rappellera » et « votre chèque est au courrier départ »).

ficultés d'apprentissage, les mêmes inhibitions que celles que j'avais connues moi-même à mes débuts.

Pour surmonter ces inhibitions, il est utile d'enjamber la barrière qui sépare l'utilisateur des concepteurs et de voir comment ces derniers ont fait pour mettre au point des logiciels aussi étranges ; nous allons examiner l'histoire de deux des plus importants d'entre eux : le traitement de texte et le tableur.

3.8.1 Le traitement de texte

L'utilisateur d'un traitement de texte sur ordinateur dispose aujourd'hui d'une grande diversité de fonctions dont presque toujours il ne connaît qu'une partie. Cette accumulation s'est faite progressivement. L'histoire de WordPerfect permet de l'illustrer : à partir de l'innovation initiale, ce produit a en effet parcouru toutes les étapes de l'évolution (Peterson [66]).

* *

En 1977 le traitement de texte est un sujet neuf. Les machines de traitement de texte (comme la Wang) sont des machines à écrire dotées d'un processeur et d'une mémoire, et la moins chère d'entre elles coûte 15 000 \$. Il existe des traitements de texte informatiques « au kilomètre » (« run-off ») : l'affichage à l'écran est hérissé de codes, il faut reformater le document avant de l'imprimer et on ne découvre son apparence qu'après l'impression.

Alan Ashton, professeur d'informatique, se lance alors pour le plaisir dans la conception d'un traitement de texte. Il avait auparavant écrit un programme de musique sur ordinateur, ce qui l'avait conduit à traiter les problèmes que pose l'affichage en temps réel.

Ashton produisit un programme qui permettait d'afficher à l'écran l'apparence de ce qui serait imprimé et de faire défiler le texte à l'écran sans interruption et non plus page à page. Il supprima la distinction entre les modes *Edit*, *Insert* et *Create* : l'utilisateur pouvait taper partout dans le document et y insérer du texte sans avoir à changer de mode.

Ce programme, amélioré et industrialisé, deviendra WordPerfect et sera commercialisé en 1980 par la société SSI (créée en 1978). Son prix était de 5 500 \$: il était économique pour une entreprise d'acheter un logiciel fonctionnant sur ordinateur plutôt qu'une machine de traitement de texte ; cependant il fallait être sûr que l'exécution de WordPerfect ne ralentirait pas l'ordinateur, utilisé alors surtout pour la comptabilité.

La première version de WordPerfect était rapide et facile à utiliser mais elle ne fonctionnait que sur les ordinateurs de Data General, sous le système d'exploitation AOS et avec une imprimante Diabolo 1650. Durant les années suivantes, une grande part du travail de programmation sera consacrée à l'adaptation de WordPerfect aux divers systèmes d'exploitation²⁶ et imprimantes du marché.

Le 12 août 1981 IBM sort le PC. Wordstar, le premier traitement de texte pour PC est lancé par Micropro au milieu de 1982. Il s'agit de la transcription à MS-DOS

26. WordPerfect comportera finalement des versions pour les mainframes d'IBM, les VAX, des machines Unix, l'Apple II, l'Amiga, l'Atari, le Macintosh, le PC etc.

d'un programme de traitement de texte sous CP/M (un des tout premiers systèmes d'exploitation pour micro-ordinateur).

* *

SSI se lance elle aussi dans la mise au point du traitement de texte sur PC. WordPerfect pour PC sera semblable à la version Data General. Toutefois sur un PC on n'a pas à se soucier d'avoir plusieurs utilisateurs simultanés. SSI introduit les notes en bas de page et le correcteur d'orthographe. WordPerfect pour PC sort le 18 novembre 1982. Il est adapté à l'imprimante Epson commercialisée par IBM. En 1983, Microsoft sort Word qui est généralement jugé inférieur à WordPerfect.

WordPerfect s'adapte aux machines MS-DOS non IBM (Victor 9000, DEC Rainbow, Tandy 2000, TI Professional etc.) et il est mis à jour pour tenir compte de l'arrivée de nouveaux périphériques (imprimantes, écrans, claviers, disques durs).

Le produit présente encore des défauts : certains pilotes d'imprimantes sont bogués, le manuel n'est pas jugé « professionnel ». Alors que les imprimantes « bêtes » accordaient la même largeur à chaque caractère, les nouvelles imprimantes « intelligentes » sont capables de calculer les intervalles convenables pour imprimer divers types de caractères. WordPerfect décide de placer les instructions pour imprimantes dans une table située hors du programme, ce qui permettra de l'adapter plus facilement à de nouvelles imprimantes. WordPerfect 3.0 sera ainsi capable de servir plus de cinquante types d'imprimantes.

SSI décide de mettre en place un centre d'appel gratuit pour les utilisateurs : cette mesure fera beaucoup pour le succès commercial du produit. Le service sera renforcé au printemps 1990 par la création d'un « hold jockey », personne qui anime les appels en attendant de la musique, des commentaires etc.

Avec WordPerfect 4.0 en 1984 le manuel est amélioré, l'installation est plus simple, le dictionnaire plus riche ; on ajoute les notes en fin de texte et le traitement des erreurs est meilleur.

Wordstar est encore le leader mais Micropro scie la branche sur laquelle il était assis en lançant Wordstar 2000 qui, étant en rupture avec l'ergonomie antérieure, déconcerte les clients les plus fidèles.

WordPerfect 4.1 sort à l'automne 1985. Il comporte la table des matières automatique, l'indexation automatique, la possibilité d'étaler les notes de bas de page sur des pages successives, la numérotation des paragraphes, le thesaurus, une vérification d'orthographe améliorée.

La part de marché de WordPerfect s'améliore. Il tire profit des erreurs de ses concurrents : Wang a décidé d'ignorer le marché du PC et continue à produire des machines à écrire informatisées, ce qui entraînera sa chute. IBM a conquis une bonne part de marché avec Displaywrite grâce à la magie de son nom mais ce produit reste inférieur à WordPerfect²⁷.

Chez Micropro, le turn-over des programmeurs est élevé, ce qui empêche l'accumulation d'expertise. Lotus, leader sur le marché du tableur, comprend mal celui du traitement de texte. Microsoft est le seul concurrent dangereux, sa maîtrise du

27. Selon Peterson, alors que SSI voyait dans la programmation un art IBM la considérait comme une *industrie* et mesurait la production selon le nombre de lignes du code source. Il en résultait des programmes de qualité médiocre (Microsoft avait fait la même observation : voir note p. 94).

système d'exploitation lui donnant un avantage stratégique tant au plan commercial qu'au plan de la conception technique.

SSI prend en 1986 le nom de WordPerfect Corporation. Mais l'arrivée de l'imprimante à laser et de l'interface graphique entraîne un changement des règles du jeu qui donne à Word l'occasion de rattraper WordPerfect. Il fallait en effet pour s'adapter aux nouvelles règles réécrire les parties du programme concernant les imprimantes et l'affichage à l'écran ; les mesures devaient être exprimées en centimètres et non plus en lignes et en espaces ; il fallait connaître les dimensions de chaque caractère dans chaque type et savoir charger les types sur les imprimantes. Alors qu'en mode texte l'ordinateur traitait un écran formé de 25 lignes de 80 caractères, soit 2000 boîtes, en mode graphique il travaillait avec 640 * 480 points, soit plus de 300 000 éléments : l'affichage à l'écran était plus lent.

Il était d'ailleurs difficile au plan stratégique de prévoir qui serait le vainqueur sur le marché de l'interface graphique pour PC : les concurrents étaient IBM avec TopView, Digital Research avec Gem et Microsoft avec Windows. Sortir une version de WordPerfect pour Windows aurait apporté un soutien à Microsoft qui, par la suite, pourrait être en mesure d'évincer WordPerfect.

WordPerfect, incapable de résoudre à la fois tous ces problèmes, se concentra d'abord sur l'imprimante à Laser. WordPerfect 5.0 ne sera prêt qu'en mai 1988.

* *

En 1987 WordPerfect a 30 % du marché devant Micropro à 16 %, IBM à 13 % et Microsoft à 11 %. Il tire un argument commercial du besoin de compatibilité entre les divers documents produits par une entreprise, voire par des entreprises différentes : les entreprises réclament un standard.

La stratégie était de produire une version pour chaque plate-forme significative, puis d'intégrer WordPerfect avec les autres produits importants sur chaque plate-forme : sur le marché du PC, avec le tableur Lotus 1-2-3 ; sur le marché du VAX, avec All-in-One etc.

Pour ne pas dérouter les utilisateurs il fallait que WordPerfect 5.0 ne fût pas trop différent des versions précédentes. On pouvait cependant y introduire discrètement les possibilités graphiques du « desktop publishing » : pour intégrer texte et graphique sur un document, il faut savoir répartir le texte autour de boîtes contenant les graphiques et savoir faire en sorte que selon les besoins ces boîtes restent en place ou au contraire se déplacent avec le texte dans le document. Les graphiques pouvant être volumineux, il fallait aussi savoir traiter de gros fichiers. MS-DOS offrant peu de possibilités graphiques, il fallait enfin écrire les logiciels nécessaires pour composer et publier les graphiques.

L'une des décisions les plus importantes fut de ne pas faire de 5.0 un produit pleinement Wysiwyg. Le Wysiwyg implique non seulement que les textes gras et soulignés, les fins de phrase et les fins de page s'affichent comme sur l'imprimé, mais aussi que les caractères soient de même style et taille et que les notes de bas de page et les graphiques soient placés au bon endroit. Ce n'était pas facile avec MS-DOS qui ne fournissait pas beaucoup d'outils pour traiter l'écran. Windows, lui, fournissait les pilotes d'écran, mais il était lent et peu fiable.

WordPerfect décida donc que l'écriture se ferait en mode texte, le mode graphique n'étant utilisé que pour visualiser la page à imprimer et y insérer les graphiques. Le

Wysiwyg complet attendrait la version 6.0.

Les autres améliorations de la version 5.0 concernent la référence automatique (« voir p. 17 » se met à jour si le contenu de la page 17 est déplacé à une autre page), le support pour 1500 caractères y compris les caractères internationaux et diacritiques, l'impression intelligente (adapter au mieux le document à l'imprimante utilisée), la fusion de documents, les listes à puces, le suivi des modifications etc.

La diversité des matériels sur le marché contraignait WordPerfect à traiter des problèmes comme les suivants : « si l'utilisateur crée un document prévu pour l'imprimante à laser, puis l'emmène chez lui et tente de l'imprimer sur un autre type d'imprimante, WordPerfect devra-t-il reformater automatiquement le document ? Sinon, que se passera-t-il si le pilote de l'imprimante du bureau n'est pas présent sur la machine à domicile ? Si le document est reformaté automatiquement, l'utilisateur en sera-t-il informé, pourra-t-il annuler le reformatage ? »

En 1987, IBM et Microsoft lancent OS/2 et Presentation Manager, concurrent de Windows. En fait, Microsoft va miser sur Windows ; OS/2 et Presentation Manager deviendront des produits IBM. Craignant qu'un succès de Windows ne donne l'avantage à Microsoft, qui connaissant bien sa propre interface graphique pourrait mieux la maîtriser, WordPerfect va donner la préférence à OS/2.

Lorsque WordPerfect 5.0 pour MS-DOS sort enfin en mai 1988 il faudra surmonter des bogues dans le programme d'installation ainsi que dans certains pilotes d'imprimante et de carte graphique. Le produit a tout de même du succès : raccordé à une imprimante à laser, le PC fournit avec WordPerfect une qualité d'impression semblable à celle du Macintosh. WordPerfect prend par ailleurs une part du marché du « desktop publishing ».

* *

La version 5.1 de WordPerfect sort à l'automne de 1989. Elle est encore sous MS-DOS mais comporte les menus déroulants et la souris. Elle apporte des améliorations dans le traitement des tableaux, la fusion de documents, la tabulation, ainsi qu'un jeu de caractères élargi. Son installation est encore plus facile.

En 1990 Microsoft offre à WordPerfect Windows 3.0 en beta test, mais WordPerfect manque de développeurs expérimentés en Windows et préfère aider au succès d'OS/2.

Cependant le 31 mai 1990 Microsoft sort Windows 3.0 et les pires craintes de WordPerfect se réalisent : beaucoup d'utilisateurs veulent Windows, même bogué, et à l'occasion ils prennent aussi Word. WordPerfect décide alors de laisser tomber OS/2 et renonce à la version 5.2 pour MS-DOS afin de ne pas retarder la version Windows.

La mise au point de celle-ci est difficile. WordPerfect pour Windows ne sort qu'en novembre 1991. Le produit est lent et comporte des bogues ; néanmoins WordPerfect se vend toujours mieux que Word. Pour pouvoir développer la version suivante, les programmeurs devront attendre de disposer du DDE (dynamic data exchange) de Microsoft.

WordPerfect for Windows 6.0 sortira en 1993. A cette date, WordPerfect détient 51 % du marché du traitement de texte pour Windows, mais Word est bien placé pour devenir le leader. À la suite de l'échec d'une tentative d'entrée en bourse, WordPerfect sera vendue à Novell en 1994. Novell revendra en 1996 ses droits sur

WordPerfect à Corel qui en est aujourd'hui propriétaire.

WordPerfect a des millions d'utilisateurs mais c'est Microsoft qui, avec Word, domine aujourd'hui le marché du traitement de texte sur PC où il est concurrencé par quelques logiciels libres (notamment OpenOffice).

3.8.2 Histoire du tableur

Dans le langage des comptables américains, le mot « spreadsheet » désignait depuis toujours une grande feuille de papier, divisée en lignes et en colonnes et utilisée pour présenter les comptes d'une entreprise. La meilleure traduction de ce mot est « tableau ».

En 1961 Richard Mattessich [57], professeur à Berkeley, développa en Fortran IV un « computerized spreadsheet » fonctionnant sur un mainframe. Ce programme évaluait automatiquement certaines cases et permettait des simulations. Il est le précurseur des « electronic spreadsheets » d'aujourd'hui, que l'on appelle en français « tableurs ».

< 1961	Les comptables utilisent des tableaux sur papier .
1961	Computerized Spreadsheet de Mattesich en Fortran IV.
1978	Création de Software Arts.
1979	VisiCalc pour l'Apple II, de Software Arts, commercialisé par VisiCorp.
1981	VisiCalc est adapté à divers systèmes, notamment au PC d'IBM.
1982	Multiplan sous MS-DOS, de Microsoft.
1983	Lotus 1-2-3 , de Lotus Corp.
1984	Excel pour le Macintosh, de Microsoft.
1985	Lotus achète Software Arts.
1987	Excel 2.0 pour le PC, de Microsoft.
1995	IBM achète Lotus. Excel est désormais leader sur le marché des tableurs.
> 2000	Gnumeric , KSpread , CALC etc. offerts en logiciel libre.

TAB. 3.1 – *Chronologie du tableur*

En 1978, Daniel Bricklin, étudiant à Harvard, devait établir des tableaux comptables pour une étude de cas sur Pepsi-Cola. Plutôt que de calculer à la main il préféra programmer « un tableau noir et une craie électroniques », selon sa propre expression. Son premier prototype, en Basic, pouvait manipuler un tableau de vingt lignes et cinq colonnes.

Bricklin se fit aider ensuite par Bob Frankston, du MIT. Celui-ci réécrivit le programme en assembleur et le condensa en 20 koctets pour qu'il puisse fonctionner sur un micro-ordinateur.

À l'automne 1978, Daniel Fylstra, ancien du MIT et rédacteur à Byte Magazine, perçut le potentiel commercial de ce produit. Il suggéra de l'adapter à l'Apple-II ainsi qu'aux systèmes HP85 et HP87.

En janvier 1979 Bricklin et Frankston créèrent Software Arts Corporation (Bricklin [11]); en mai 1979, la société Personal Software de Fylstra, nommée plus tard VisiCorp, lança la commercialisation de VisiCalc (cette appellation condense l'expression « Visible Calculator »).

VisiCalc était vendu 100 \$. Il avait déjà l'allure des tableurs d'aujourd'hui : les évolutions ultérieures les plus visibles porteront sur l'adjonction de possibilités graphiques ainsi que sur l'utilisation de la souris. Le succès ne fut pas immédiat mais néanmoins rapide. Dès juillet 1979 Ben Rosen publia une analyse prophétique (Rosen [72]). Jusqu'alors seuls des hobbyistes, qui savaient programmer, pouvaient utiliser le micro-ordinateur : *VisiCalc était le premier logiciel qui permettait d'utiliser un ordinateur sans avoir à programmer*. Il contribuera fortement à la pénétration du micro-ordinateur dans les entreprises. Des versions furent produites pour diverses plates-formes, notamment pour le PC d'IBM dès son lancement en 1981.

Cependant les promoteurs de VisiCalc, empêtrés dans un conflit entre Software Arts et VisiCorp, ne surent pas faire évoluer leur produit assez rapidement.

* *

Mitch Kapor avait travaillé pour Personal Software en 1980 et proposé un produit que les dirigeants de VisiCorp refusèrent parce qu'ils l'estimaient trop limité. Il créa Lotus Development Corporation en 1982 et lança Lotus 1-2-3 en 1983. Lotus 1-2-3 pouvait être adapté plus facilement que VisiCalc à divers systèmes d'exploitation et apportait des possibilités nouvelles : graphiques, bases de données, dénomination des cellules, macros. Il devint rapidement le nouveau tableur standard.

En 1985, Lotus achètera Software Arts et arrêtera la commercialisation de VisiCalc.

* *

Microsoft s'était intéressé au tableur dès 1980. En 1982, il lance Multiplan pour le PC. Ce produit n'aura pas grand succès aux États-Unis où Lotus 1-2-3 est dominant. Par contre il sera largement utilisé ailleurs et il ouvrira la voie aux autres applications produites par Microsoft.

En 1984, Microsoft sort Excel pour le Macintosh. Le produit tire parti de l'interface graphique, des menus déroulants, de la souris : tout cela le rend plus commode que Multiplan. Tout comme VisiCalc avait contribué au succès du PC, Excel contribuera au succès du Macintosh.

En 1987 sort Excel pour PC : ce sera l'application phare de Windows. La principale amélioration par rapport à Lotus 1-2-3 est la possibilité de programmer de véritables applications avec des macro-instructions (l'utilisateur individuel ne s'en servira cependant pas beaucoup).

En 1987, Microsoft Works inaugure la famille des « office suites » en offrant dans un même package le tableur, le traitement de texte et le logiciel graphique. Excel sera jusqu'en 1992 le seul tableur disponible sous Windows.

A la fin des années 80, Lotus et Microsoft dominent le marché malgré l'arrivée de nombreux autres tableurs (Quattro de Borland, SuperCalc de Computer Associates etc.) La concurrence est vive et suscite des batailles juridiques : procès entre Lotus et Software Arts, gagné par Lotus en 1993 ; procès entre Lotus et Mosaic d'une part, Paperback de l'autre, gagnés par Lotus en 1987 : Lotus gagnera toutes ses batailles juridiques mais perdra contre Microsoft la bataille pour la domination du marché.

En 1990, un juge lèvera le copyright de Lotus sur l'interface utilisateur, estimant que « rien dans cette interface n'était inséparable de l'idée du tableur ». En 1995,

IBM achète Lotus, alors qu'Excel domine le marché.

Plus de 20 tableurs sont aujourd'hui offerts dans le monde du logiciel libre (« open source »). Gnumeric est souvent distribué en même temps que Linux. Parmi les autres tableurs, les plus connus sont KSpread et CALC.

* *

Si VisiCalc présente déjà un aspect qui nous est familier, le tableur s'est progressivement enrichi. L'adresse des cellules, d'abord notée selon le format L1C1 (R1C1 pour les anglophones), a pu ensuite s'écrire sous la notation condensée A1. L'existence de deux types d'adresse (adresses relatives, adresses absolues de type \$A\$1) a allégé la programmation. L'introduction des feuilles et des liens a permis de doter le tableur d'une troisième dimension (la feuille s'ajoutant à la ligne et à la colonne), voire d'un nombre quelconque de dimensions si on relie plusieurs tableurs.

Lotus 1-2-3 a apporté les outils graphiques qui facilitent la visualisation des résultats. Les macros (également introduites par Lotus 1-2-3 en 1983, puis perfectionnées par Microsoft) permettent de programmer des applications sur le tableur. La souris (à partir de 1984 avec Excel sur le Macintosh) a facilité la sélection des plages de cellules et la dissémination des formules par glissement du pointeur. Le solveur (introduit en 1990 par Frontline) permet de résoudre des problèmes de calcul numérique, d'économétrie, de recherche opérationnelle etc.

* *

Dans l'entreprise, le tableur est utilisé pour les simulations, les calculs sur les séries chronologiques, la comptabilité, la préparation de rapports ou de déclarations fiscales. Des fonctions simples sont utilisées de façon répétitive pour faire des additions et calculer des moyennes. Les utilisations scientifiques (calcul numérique, visualisation de statistiques, résolution d'équations différentielles) sont plus compliquées et moins répétitives.

Pour l'utilisateur de base, le tableur n'est que la fusion électronique du papier, du crayon et de la calculatrice. Il n'a généralement pas été formé à s'en servir et il est le plus souvent peu conscient des conséquences que risque d'avoir une erreur. C'est un expert dans son métier et il ne se considère pas comme un programmeur. Il veut traiter rapidement son problème et ne souhaite ni recevoir les conseils d'un informaticien, ni partager son expertise avec lui. Il est en pratique impossible de lui imposer des méthodes strictes de programmation ou de vérification. Son développement progresse par essais et erreurs : il construit un premier prototype puis le modifie jusqu'à ce qu'il réponde à ses besoins. Pressé d'arriver à ses fins, il néglige de documenter son programme. Celui-ci ne pourra pratiquement jamais être réutilisé par quelqu'un d'autre et son créateur lui-même aura du mal à le faire évoluer ou à le maintenir.

Les dirigeants de l'entreprise, pour leur part, n'utilisent généralement pas le tableur mais sont destinataires de tableaux de bord et autres « reportings » imprimés sur papier mais construits sur des tableurs. Ils lisent ces tableaux comme s'ils provenaient d'un traitement de texte, sans concevoir les calculs dont ils résultent.

L'entreprise donne donc au tableur un rôle ambigu : c'est un outil de travail commode mis à la disposition de tous, mais générateur d'erreurs et les programmes sont difficiles à entretenir. Vu l'importance prise par le tableur dans le système d'infor-

mation de l'entreprise, il est utile de repérer les erreurs et d'utiliser les méthodes qui permettent de les éviter (O'Beirne [62]).

* *

Une enquête a montré que la majorité des tableurs contenaient des cellules fausses, et qu'en moyenne 3 % des cellules d'un tableur étaient erronées (Clermont [16] et Panko [63]). Certaines erreurs ont eu des conséquences graves :

1) Les données utilisées pour passer une commande sont désuètes : 30 000 pièces à 4 \$ sont commandées au lieu de 1 500, ce qui entraîne une perte de 114 000 \$.

2) Dans une étude prévisionnelle, les sommes en dollars sont arrondies à l'unité : le multiplicateur de 1,06 qui représente l'effet de l'inflation est alors arrondi à 1. Le marché d'un produit nouveau est sous-estimé de 36 000 000 \$.

3) Un tableur a été programmé par une personne qui a quitté l'entreprise et n'a pas laissé de documentation : le taux d'actualisation utilisé pour calculer la valeur actuelle nette des projets est resté à 8 % entre 1973 et 1981 alors qu'il aurait dû être porté à 20 %, d'où des erreurs dans le choix des investissements.

4) Dans la réponse à un appel d'offre une addition est inexacte (des rubriques ajoutées à la liste n'ont pas été prises en compte) : l'entreprise sous-estime de 250 000 \$ le coût du projet et elle intente par la suite un procès à Lotus.

5) Un comptable fait une erreur de signe lors de la saisie d'un compte de 1,2 milliards de dollars : l'entreprise prévoit un profit de 2,3 milliards et annonce une distribution de dividendes. Finalement elle constate une perte de 100 millions de dollars et doit déclarer qu'aucun dividende ne sera distribué, ce qui dégrade bien sûr son image auprès des actionnaires.

6) En 1992, 10 % des tableurs envoyés aux inspecteurs des impôts britanniques pour le calcul de la TVA contenaient des erreurs. Il en est résulté une perte de recettes de 5 000 000 £.

Certaines des erreurs relevées dans l'utilisation du tableur sont de celles que l'on peut rencontrer dans d'autres démarches : la représentation du monde réel par un modèle peut être non pertinente ou dégradée par des défauts du raisonnement (additionner des données hétéroclites, des ratios etc.) D'autres erreurs sont commises lors de la programmation : on confond référence relative et référence absolue, on se trompe dans la syntaxe des formules (notamment dans l'utilisation des parenthèses) : ces erreurs, qui révèlent une mauvaise compréhension du fonctionnement du tableur, sont ensuite disséminées par la réplique des cellules. Presque toujours enfin on néglige de documenter le tableur, ce qui rendra sa maintenance difficile surtout si l'on a programmé des macros.

Puis viennent les erreurs commises lors de l'utilisation : erreurs de saisie, erreur dans la correction d'une formule, remplacement *ad hoc* d'une formule par une constante qui, restant dans le tableau, polluera les calculs ultérieurs, mauvaise définition de l'aire couverte par une formule, absence de mise à jour de cette aire lorsque des lignes sont ajoutées.

Utiliser les macros comme des boîtes noires (par exemple pour le calcul du taux de rentabilité d'un projet) peut interdire de traiter convenablement le cas particulier que l'on étudie. Il arrive aussi que le solveur converge mal : le prendre au pied de la lettre donne un résultat aberrant.

L'édition des tableaux sur papier est l'occasion d'erreurs de présentation : tableaux sans titre, sans intitulé de ligne et de colonne, sans nom d'auteur, sans date ni indication de la période représentée ; erreurs sur les unités de mesure (euro à la place de dollar, millions à la place de milliards).

On relève enfin des erreurs dans les graphiques : représenter une série chronologique par un histogramme, ou pis par un fromage au lieu d'une courbe ; utiliser une courbe au lieu d'un histogramme pour une distribution ; quand on utilise conjointement deux échelles, mal représenter les évolutions relatives etc.

* *

Leçons tirées de l'histoire de la bureautique

Pour étudier l'histoire du tableur ou du traitement de texte, il faut dépouiller beaucoup de documents. Deux faits sautent alors aux yeux : l'un concerne la documentation elle-même, l'autre concerne les inventeurs.

La plupart des documents accordent une grande place à l'identité des innovateurs et à l'évolution du marché : *qui* a inventé le produit, et quand ; *quand*, et avec qui, il a créé son entreprise ; *quand*, et comment, s'est manifestée la concurrence ; *comment* ont évolué les parts de marché ; *comment* se sont réglés les conflits.

Les indications sur la nature du produit partent presque toutes du point de vue de l'utilisateur car il s'agit d'expliquer le succès commercial. Si l'idée qui a guidé la conception est indiquée, on ne trouve pas grand-chose sur les choix fonctionnels ou d'architecture ni sur les compromis auxquels le réalisateur a été contraint : le passage de l'idée au produit s'est déroulé dans une boîte noire que la documentation n'entrouvre pas.

Il est vrai qu'il est difficile de décrire ce passage, qui est délicat et laisse le plus souvent peu de traces documentaires. Cela ne donne que plus de valeur aux ouvrages qui, comme celui de Tracy Kidder [50], décrivent en détail le cheminement d'une réalisation technique.

Dans la plupart des cas, l'inventeur un homme jeune récemment sorti de l'université ou même encore un étudiant. Il est au courant des techniques récentes et, comme il n'a aucune expérience, son esprit est libre de préjugés. Il imagine le nouveau produit en partant de ses propres besoins.

Beaucoup des produits ainsi conçus n'aboutissent à rien de durable, et ceux-là seront oubliés : l'histoire ne garde en mémoire que les produits réussis. Notre étudiant, si son produit rencontre le succès, devient un entrepreneur : il découvre les soucis du recrutement, de la commercialisation, des partenariats, de la gestion, du financement. Pour défendre son territoire, il s'englué dans des procédures. Il perd ainsi sa capacité à innover et d'ailleurs son savoir technique, naguère pointu, devient banal. Il laisse passer des occasions qu'il ne perçoit plus et se fait doubler par d'autres personnes tout juste sorties de l'université. Bientôt la survie de son entreprise est menacée...

Le système est innovant parce que le flux des jeunes innovateurs ne tarit pas. Par contre si l'on considère les individus, le système paraît stérilisant : tout se passe comme si le fait de mettre une idée sur le marché inhibait la possibilité d'en avoir une autre. Quant aux entreprises, elles naissent, croissent et meurent ou se font acheter en quelques années.

Microsoft est l'exception qui confirme la règle car si Bill Gates innove c'est en entrepreneur, en organisateur plus qu'en technicien : il reprend les produits inventés par d'autres et les articule en système avec un sens aigu des économies d'envergure et du fonctionnement du marché.

3.9 Esquisse de prospective

Rappelons ce qu'était l'informatique dans les entreprises en 1988. Le Web n'existait pas (il a été inventé en 1991) et on ne connaissait guère l'Internet. L'informatique ne traitait pratiquement que des données structurées. La messagerie, les workflows, la documentation électronique existaient sur de gros systèmes mais seuls des informaticiens les utilisaient car ils étaient peu ergonomiques. Les réseaux locaux étaient rares (ils ne se multiplieront qu'à partir de 1989). Le téléphone mobile était lourd (il n'équipait pratiquement que des automobiles) et coûteux.

Les micro-ordinateurs étaient chers : un PC à 16 MHz, avec un disque dur de 80 Mo et une RAM de 2 Mo, coûtait plus de 10 000 € TTC aux prix de 2005. L'ergonomie était rudimentaire à l'exception de celle des Macintosh (Windows 95 date, comme son nom l'indique, de 1995).

* *

Qu'avons-nous acquis depuis 1988 ? L'informatique de communication s'est développée : la messagerie est entrée dans les mœurs ainsi que la documentation électronique (Intranet et Web). L'ordinateur a acquis l'ubiquité : la machine dont chacun dispose permet d'accéder *via* le réseau à des ressources informatiques (mémoire, puissance de traitement) dont la localisation physique importe peu. Le travail assisté par ordinateur s'est généralisé : l'automate soulage le travail mental de l'être humain en l'assistant dans les tâches de classement, recherche, transcription et calcul.

Le système d'information définit le langage de l'entreprise dont il incorpore la sémantique. Il outille son action, ses processus de travail. Il équipe progressivement tout le personnel. La maîtrise intellectuelle et pratique du système d'information suppose que l'on sache urbaniser sa structure d'ensemble et modéliser chacun des processus de production. L'automatisation (partielle) des processus doit être associée à une vigilance qui permettra de trouver des solutions raisonnables aux cas non prévus lors de la programmation de l'automate.

* *

Plaçons nous par l'imagination en 2015. Les composants essentiels de l'informatique communicante existent déjà aujourd'hui (processeurs, mémoires, réseaux) ; le changement à cette échéance réside donc moins dans la nouveauté des composants (dont toutefois les performances auront depuis 2005 été multipliées par $2^{10/1,5} \approx 100$ selon la loi de Moore) que dans la transformation des interfaces et protocoles permettant de les commander et de les faire communiquer, donc dans une évolution des conditions de leur utilisation.

En 2015, en raisonnant par extrapolation tendancielle, téléphone portable et PC se seront miniaturisés et ils auront fusionné. Les réseaux permettront une connexion permanente à haut débit en mode paquet selon le protocole TCP/IP de l'Internet, y

compris pendant les déplacements. Les personnes, les objets seront ainsi équipés de ressources informatiques et de communication intégrées dans des composants minuscules. Les thèmes principaux de l'offre auront les noms suivants : « communicateurs personnels », « containers d'information », « télédiagnostic des équipements » etc. De nombreux « téléservices » pourront les utiliser.

L'ordinateur « wearable », portable au sens où l'on dit que l'on « porte » des vêtements, sera devenu discret et pratique. Les personnes disposeront ainsi de fonctions informatiques et de communication qui équiperont non plus le bureau, mais le corps lui-même²⁸ ; chaleur et mouvements du corps fourniront l'énergie nécessaire. L'écran sera incorporé aux lunettes, ou bien il s'affichera sur une feuille que l'on peut déplier comme un journal. Processeur, mémoire et disque dur seront intégrés dans un boîtier qui servira de palm top et de clavier. Le son sera fourni par un walkman ou par un composant inséré dans l'oreille. Les commandes seront saisies par reconnaissance vocale ou par clavier. Les diverses parties de l'équipement communiqueront par câble ou ondes à courte portée.

L'équipement personnel reçoit et envoie alors des messages écrits et vocaux en temps réel. L'utilisateur peut consulter des ressources, recevoir des alarmes etc. : l'équipement assiste sa mémoire, dans la continuité des services que rend aujourd'hui l'agenda sur papier, en l'enrichissant par l'accès à des ressources encyclopédiques et des moteurs de recherche.

Ainsi l'ubiquité logique (disponibilité des données et des outils de traitement quel que soit l'endroit où l'on se trouve) sera absolue²⁹, sous la seule réserve du contrôle d'accès. L'ubiquité physique restera, elle, limitée : si notre voix et notre image pourront se trouver simultanément en divers points du monde, il n'en sera pas de même de notre corps (qui pourra toutefois être représenté par son image « virtuelle » à trois dimensions (Quéau [68])).

Les personnes qui veulent communiquer avec l'utilisateur peuvent lui être présentées par leur « carte de visite » comportant une photographie (enrichissement de l'identification d'appel), et l'utilisateur a le choix entre communication synchrone et asynchrone.

La personne équipée qui se déplace dans un environnement d'objets communicants reçoit les signaux émis par ces objets et les interprète. Elle peut aussi recevoir les signaux émis par les équipements des autres personnes (identifier amis et relations dans une foule etc.)

* *

Les objets eux-mêmes sont munis de ressources informatiques, facilitant la « traçabilité » des biens de consommation (origine, composition chimique et fraîcheur des produits alimentaires, identification des fournisseurs ayant participé à l'élaboration d'un produit composite etc.) La traçabilité des produits, notamment alimentaires, constituera un avantage compétitif et se traduira soit par l'acceptation de prix plus élevés, soit par l'élimination progressive des produits non « tracés ». Des étiquettes

28. Certaines recherches visent à équiper non seulement le vêtement, mais le corps ; il s'agit de compenser des handicaps (cécité, surdité, troubles de l'olfaction), mais aussi de multiplier les capacités sensorielles en équipant le corps de capteurs plus sensibles que les capteurs naturels.

29. L'accès à la ressource informatique ne sera plus conditionné par la proximité avec un ordinateur.

rayonnantes permettront de les identifier, puis de trouver sur le Web les informations complémentaires.

L'appartement est truffé d'objets communicants aux fonctions diverses. Il est équipé d'un ordinateur central relié au monde par des accès à haut débit (les accès ADSL et les paraboles pour satellites en sont une préfiguration) qui constitue le centre du réseau des objets communicants, organise les fonctions informatiques, audiovisuelles et télécoms du ménage, pilote le chauffage, l'éclairage, l'arrosage du jardin etc. selon les consignes fournies par l'utilisateur.

Le terme « ordinateur » ne désigne pas ici une machine, mais un ensemble de fonctions résidant sur des machines diverses, y compris sur des machines situées hors de l'appartement mais fonctionnant sous le contrôle du ménage. L'utilisateur ne dispose plus comme aujourd'hui de plusieurs ordinateurs (un au bureau, un au domicile, un palm top, un ordinateur portable, et en outre un téléphone filaire et un téléphone portable) entre lesquels il doit recopier les données : il dispose d'*une ressource informatique*, localisée sur des serveurs dont l'emplacement géographique lui indiffère et entre lesquels traitements et données se répartissent. Il accède à cette ressource par des interfaces diverses sans que cela altère l'unité de celle-ci. À la fois informatique et téléphonique, cette ressource gère les messageries écrites et vocales, garde mémoire des communications vocales ou des conversations. Elle est connectée en permanence au Web sur lequel elle réalise des missions de recherche et de tri.

* *

Dans ce scénario, *rien ne dépasse les possibilités de la technique actuelle* qui sont déjà immenses, beaucoup plus importantes qu'on ne le croit³⁰. Ce qui est nouveau, c'est l'intégration des applications qui permet de supprimer les ressaisies ; c'est aussi un filtrage sélectif permettant de trier sur le Web l'utile de l'accessoire. L'utilisateur peut ainsi être assisté ou éclairé dans toutes ses actions : la logique de l'*assisté par ordinateur* déploie ses implications. L'offre est commode, pratique, une haute complexité technique étant masquée par la facilité de l'usage.

Cette évolution, d'ores et déjà en préparation dans les centres de recherche, chez les fournisseurs et les opérateurs télécoms, comporte des risques évidents :

1) *risque de dépendance* envers un système qui assisterait l'utilisateur en permanence : un nouveau savoir-vivre, une nouvelle hygiène, sont ici nécessaires. De même qu'il est aujourd'hui déconseillé de regarder la télévision sans discontinuer, il sera déconseillé d'utiliser en permanence les équipements informatiques.

Un système qui permet de recevoir en temps réel alarmes, messages et communications, qui permet à d'autres personnes de vous localiser, qui peut à tout moment accéder à des ressources (images, données, textes, sons, jeux), peut être oppressant. Il doit comporter divers niveaux de veille, de l'arrêt total à l'ouverture totale, en passant par un blocage sélectif des communications ne laissant passer que certaines alarmes, certains messages etc. Il faudra savoir se débrancher et utiliser la communication asynchrone (déjà, aujourd'hui, il faut savoir débrancher le téléphone mobile).

2) *sécurité* : l'importance prise par l'automatisation a pour corollaire l'obligation

30. Par exemple l'ensemble des conversations auxquelles participe un même individu durant sa vie, des cours qu'il suit, des émissions de radio qu'il écoute, pourrait une fois transcrit en mode caractère être horodaté, documenté et stocké sur un seul DVD...

de *contrôler les automates* car personne ne doit faire entièrement confiance à des automatismes pour sa vie courante. La communication entre automates, les actions qu'ils déclenchent doivent pouvoir être traçables et contrôlables ; l'utilisateur doit disposer d'interfaces commodes pour les paramétrer ; elles doivent être sécurisées pour éviter les fausses manœuvres. La protection de la vie privée suppose enfin que les automates soient protégés contre toute tentative d'indiscrétion.

3) *risque de rejet* : le monde dans lequel vivra l'utilisateur est différent de celui que nous connaissons aujourd'hui : les appareils seront plus discrets, mais les fonctionnalités seront omniprésentes. Cette évolution suscitera des réactions de rejet comme l'ont fait en d'autres temps le téléphone, l'ordinateur, le Minitel, voire les équipements électroménagers (machine à laver, aspirateur etc.)

Il faut être averti de ces risques, mais il ne faut pas les exagérer : parmi les périls auxquels l'humanité est confrontée, ils ne sont pas les plus graves (voir la note page 12), et ils doivent être mis en balance avec les avantages qu'apporte l'informatique.

* *

Soyons optimistes : les entreprises se seront dotées de compétences en ingénierie d'affaires et il leur sera devenu plus facile d'assurer l'interopérabilité des systèmes d'information. La plupart des produits offerts au consommateur seront des « packages » auxquels plusieurs entreprises auront contribué dans le cadre d'accords de partenariat. Les systèmes d'information seront mieux maîtrisés, qu'il s'agisse d'urbanisation ou de modélisation : les entreprises auront surmonté les difficultés sociologiques et intellectuelles qu'elles rencontrent aujourd'hui, et disposeront de méthodes et outils dans la lignée ouverte par UML. On ne parlera plus d'applications, mais de processus et de composants. Les problèmes de normalisation auront été surmontés...



FIG. 3.2 – La revue qui a éveillé les premières vocations

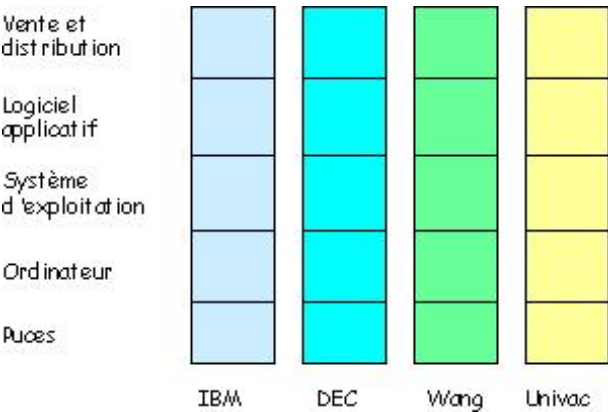


FIG. 3.3 – Partage du marché en 1980

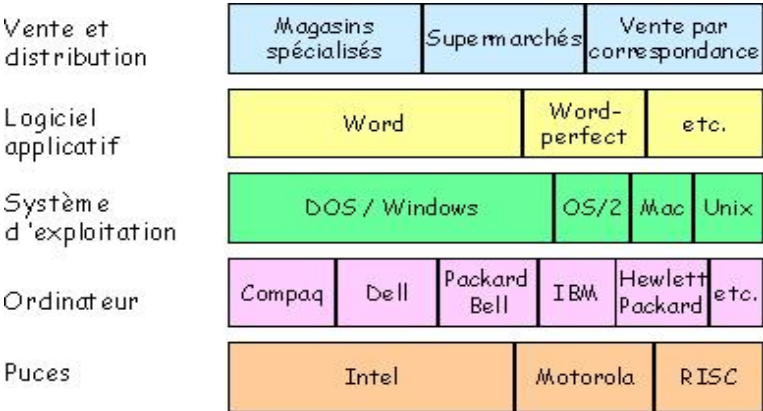


FIG. 3.4 – Partage du marché en 1990

Bibliographie

- [1] *Perspectives of the Founding Fathers*, Winter Simulation Conference, 1992.
- [2] Masahiko Aoki. Information and governance in the silicon valley model, 1999.
- [3] Jacques Arsac. *Préceptes pour programmer*. Dunod, 1991.
- [4] David S. Bennahum. *Extra Life*. Basic Books, 1998.
- [5] Laurent Bloch. *Initiation à la programmation avec Scheme*. Technip, 2001.
- [6] Laurent Bloch. *Les systèmes d'exploitation des ordinateurs : histoire, fonctionnement, enjeux*. Vuibert, 2003.
Laurent Bloch a publié ce livre à l'adresse :
<http://www.laurent-bloch.org/Livre-Systeme/index.html>.
- [7] Maurice Blondel. *L'Action*. Félix Alcan, 1893.
L'œuvre maîtresse du philosophe qui, fait exceptionnel, a mis l'*action* au point de départ de sa pensée.
- [8] N. Bourbaki. *Éléments de mathématique, Théorie des ensembles*. Hermann, 1966.
- [9] Isabelle Boydens. *Informatique, normes et temps*. Bruylant, 1999.
À partir du cas particulier de la base de données de la sécurité sociale belge, Isabelle Boydens déploie une analyse précise des problèmes que pose la pratique des bases de données, et elle propose des solutions. Voir le commentaire page 64.
- [10] Philippe Breton. *Une histoire de l'informatique*. La Découverte, 1987.
- [11] Dan Bricklin. Software arts and visicalc.
www.j-walk.com/ss/history/spreadsh.htm.
- [12] Jean-Pierre Brulé. *L'informatique malade de l'Etat*. Les Belles Lettres, 1993.
- [13] René Carmille. *La mécanographie dans les administrations*. Recueil Sirey, 1942.
- [14] Paul Carroll. *Big Blues, The Unmaking of IBM*. Crown.
Une histoire détaillée de la chute de la maison IBM dans les années 90... mais IBM s'est redressée par la suite, ce qui infirme quelque peu l'analyse.
- [15] Paul E. Ceruzzi. *A History of Modern Computing*. MIT, 1998.
- [16] Markus Clermont. *A Scalable Approach to Spreadsheet Visualization*. PhD thesis, Universität Klagenfurt, mars 2003.
- [17] Robert P. Colwell. *The Pentium Chronicles*. Wiley & Sons, 2006.
- [18] Ferdinand de Saussure. *Cours de linguistique générale*. Payot, Paris, 1916.
- [19] Alexis de Tocqueville. *De la démocratie en Amérique*, 1835.
- [20] Vincent Desportes. *L'Amérique en armes*. Economica, 2002.
- [21] Gérard Dréan. Apple, sun et... ibm. *Le Monde Informatique*, 22 mars 1996.
- [22] Robert Escarpit. *L'information et la communication. Théorie générale*. Hachette, Paris, 1991.

- [23] Bertrand Russell et Alfred Whitehead. *Principia Mathematica*, 1913.
- [24] O. J. Dahl, E. W. Dijkstra et C. A. R. Hoare. *Structured Programming*. Academic Press, 1972.
- [25] Terry Winograd et Fernando Flores. *Understanding computers and cognition*. Addison-Wesley, 1989.
- [26] Harold Abelson et Gerald Jay Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, 1996.
Une des meilleures introductions qui soit à l'art de la programmation.
- [27] Guillaume Lecointre et Hervé Le Guyader. *Classification phylogénétique du vivant*. Belin, 2001.
- [28] Grady Booch, James Rumbaugh et Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1998.
- [29] Audris Mockus, Roy Fielding et James Herbsleb. « case study of open source software program movement: The apache server », 2000.
- [30] Richard Stiller et Jos Marlowe. *Asunder*. Briarwood, 1999.
- [31] Jean Tirole et Josh Lerner. The simple economics of open source. *Journal of Industrial Economics*, (52), juillet 2001.
- [32] S. Syme et L. J. Camp. Code as governance, the governance as code. *Social Science Research Network Electronic Paper Collection*, avril 2001.
- [33] Philippe Penny et Michel Volle. La téléinformatique dans l'entreprise. *La Recherche*, juin.
- [34] Elliotte Rusty Harold et W. Scott Means. *XML in a Nutshell*. O'Reilly, 2004.
- [35] Michel Foucault. *Les mots et les choses*. Gallimard, 1966.
- [36] Bill Gates. An open letter to lobbyists. *Computer Notes*, février 1976.
- [37] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme. *Monatshefte für Mathematik und Physik*, (38), 1931.
- [38] Michel Gensollen. Économie des biens informationnels. In « *Autour du Libre* », *colloque de l'Institut National des Télécommunications*, Evry, mai 2002.
- [39] Bertrand Gille. *Histoire des techniques*. Gallimard La Pléiade, Paris, 1978.
- [40] Antonio Gramsci. *Note sul Machiavelli, Americanismo e fordismo*. Istituto Gramsci, 1975.
- [41] Andrew S. Grove. *Only the Paranoid survive*. Doubleday, 1996.
- [42] J. S. Henry. The impact of lotus notes on productivity. Technical report, 1992.
- [43] Michael Hiltzik. *Dealers of Lightning - Xerox PARC and the Dawn of the Computer Age*. Harper Business, 1999.
Histoire vivante d'un centre de recherche où furent mises au point, dans les années 70, plusieurs innovations essentielles.
- [44] Andrew Hodges. *Alan Turing: The Enigma*. Walker & Company, 2000.
- [45] Watts S. Humphrey. Software unbundling : A personal perspective. *IEEE Annals of the History of Computing*, janvier-mars 2002.
- [46] Georges Ifrah. *Histoire universelle des chiffres*. Robert Laffont, 1994.
- [47] Jean-Marc Jancovici. *L'avenir climatique*. Seuil, 2002.
- [48] Antoine-Henri Jomini. *Précis de l'art de la guerre*, 1838.
- [49] Alan C. Kay. The early history of smalltalk. In *History of Programming Languages II*. Addison-Wesley, 1996.

De la découverte de Simula à la conception de Smalltalk, l'invention du premier langage de programmation orienté objet.

- [50] Tracy Kidder. *The Soul of a New Machine*. Atlantic-Little, Brown, 1981.
Le meilleur récit que je connaisse sur la conception et la construction d'un ordinateur.
Voir le commentaire page 31.
- [51] Donald E. Knuth. *The Art of Computer Programming*. Addison Wesley, 1997.
- [52] André Lalande. *Vocabulaire technique et critique de la philosophie*. PUF, 1976.
- [53] Steven Levy. *Hackers*. Delta Publishing, 1994.
- [54] Robert Ligonnière. *Préhistoire et histoire des ordinateurs*. Robert Laffont, 1987.
- [55] Pierre Lévy. *L'intelligence collective et ses objets*. La Découverte, 1994.
- [56] Michael Malone. *The Microprocessor, a Bibliography*. Springer-Verlag, 1995.
- [57] Richard Mattessich. Spreadsheet: Its first computerization (1961-1964).
www.bricklin.com/history/sai.htm.
- [58] Matthew Flatt et Shriram Krishnamurthi Matthias Felleisen, Robert Bruce Findler. *How to Design Programs*. MIT Press, 2001.
- [59] Marshall McLuhan. *Understanding Media : The Extensions of Man*. MIT Press, 1964.
- [60] Gordon E. Moore. Cramming more components into integrated circuits. *Electronics*, avril 1965.
- [61] Pierre Mounier-Kuhn. *L'Informatique en France de la Deuxième Guerre mondiale au Plan Calcul : science, industrie, politiques publiques*. PhD thesis, CNAM, 1999.
- [62] Patrick O'Beirne. *Spreadsheet Check and Control*. Systems Publishing, 2003.
- [63] Ray Panko. Spreadsheet research.
panko.cba.hawaii.edu/SSR/home.htm.
- [64] Richard E. Pattis. *Karel the Robot*. Wiley, 1995.
- [65] Linda Dailey Paulson. Wanted: More network-security graduates and research. *Computer*, février 2002.
- [66] Pete Peterson. *Almost Perfect*. Prima Publishing, 1994.
- [67] Karl Popper. *Objective Knowledge*. Oxford University Press, 1979.
- [68] Philippe Quéau. *Le virtuel : Vertus et Vertiges*. Champ Vallon, 1993.
- [69] Eric S. Raymond. The cathedral and the bazaar.
www.catb.org/~esr/writings/cathedral-bazaar/.
- [70] Eric S. Raymond. *The Art of Unix Programming*. Addison Wesley, 2004.
Description très vivante du monde de la programmation autour d'Unix, de ses intentions, de son histoire, des problèmes qu'il rencontre, des outils dont il dispose.
- [71] D. Rehr. The typewriter. *Popular Mechanics*, août 1996.
- [72] Benjamin M. Rosen. Visicalc : Breaking the personal computer software bottleneck. *Morgan Stanley Electronics Letter*, 11 juillet 1979.
- [73] Jean E. Sammet. *Programming Languages : History and Fundamentals*. Prentice-Hall, 1969.
- [74] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, juillet - octobre 1948.
- [75] Claude E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1963.

- [76] Michael Sipser. *Introduction to the Theory of Computation*. PWS, 1997.
- [77] D. Sjobbema. *Ils ont inventé l'électronique*. Publitronic/Elektor, 1999.
- [78] Richard Stallman. The gnu operating system and the free software movement. In Sam Ockman et Mark Stone Chris DiBona, editor, *Open Sources: Voices from the Open Source Revolution*. O'Reilly, 1999.
- [79] Bjarne Stroustrup. *The Design and Evolution of C++*. Addison Wesley, 1999.
- [80] Andrew Tanenbaum. *Operating Systems, Design and Implementation*. Prentice-Hall, 1987.
- [81] Andrew Tanenbaum. *Computer Networks*. Prentice-Hall, 1989.
- [82] Jean-Michel Truong. *Totalement inhumaine*. Les empêcheurs de penser en rond, 2002.
Une illustration d'un fantôme.
- [83] Alan M. Turing. Computing machinery and intelligence. *Mind*, (59), 1950.
- [84] Paul Virilio. *La bombe informatique*. Galilée, 1998.
- [85] Michel Volle. *Le métier de statisticien*. Economica, deuxième édition, 1984.
- [86] Michel Volle. *Analyse des données*. Economica, quatrième édition, 1997.
- [87] Carl von Clausewitz. *Vom Kriege*, 1832.
- [88] John von Neumann. *First Draft of a Report on the EDVAC*. Technical report, Moore School of Electrical Engineering, University of Pennsylvania, 1945.
- [89] John von Neumann. *The Computer and the Brain*. Yale Nota Bene, 2000.

Index

- 3Com, 90
- Shockley Robert, 30
- Adobe, 97
- Akers John, 96
- Alcatel, 44
- Aldus, 97
- Algol, 102
- All-in-One, 115
- Allen Paul, 89, 91, 92
- Altair, 33
- AltaVista, 78
- Alto, 93
- Amazon, 101
- AMD, 34
- Aoki Masahiko, 33
- AOL, 97
- AOS, 113
- Apache, 109
- Apple, 33, 90, 93, 95–98
- Apple II, 33, 94
- Appletalk, 112
- APT, 102
- APU (Automate Programmable doué d’Ubiquité), 15
- Arcome, 112
- Ashton & Tate, 96
- Ashton Alan, 113
- AT&T, 30
- Augustin (Saint), 75
- Backus John, 102
- Bardeen John, 30
- Basic, 102
- Bell Labs, 30, 103
- Bennahum David, 87
- Bernard Claude, 112
- Berners-Lee Tim, 96, 100
- Bezos Jeffrey, 101
- BIPE, 111
- Blankenbaker John, 86
- Bloch Laurent, 9
- Blondel Maurice, 25
- Borland, 118
- Bourbaki N., 17
- Boydens Isabelle, 64
- Brattain Walter, 30
- Bricklin Daniel, 117
- Brulé Jean-Pierre, 33
- Bull, 54, 88
- Busicom, 33
- C, 102, 103
- C++, 93, 103
- Caillies, 54
- CALC, 119
- Camp L. J., 110
- Carroll Paul, 94
- Cary Frank, 94
- Census, 54
- Cisco, 90, 110
- Clark James, 93
- Clermont Markus, 120
- CNET (Centre National d’Études des Télécommunications), 111, 112
- Cobol, 102
- Commodore, 94
- Compaq, 95–97, 104, 105
- Compuserve, 96, 97
- Computer Associates, 118
- Corel, 117
- CP/M, 114
- Data General, 113, 114
- Datawarehouse, 101
- Davidoff Monte, 91, 92
- dBase, 95, 103
- DDE (dynamic data exchange), 116
- de Saussure Ferdinand, 25, 29
- de Tocqueville Alexis, 49
- DEC, 96, 104, 114
- Dell, 101, 105
- Department of Labor, 39

- Diabolo, 113
- Digital, 97
- Dijkstra Edsger, 66
- Displaywrite, 114
- DMCA (Digital Millenium Copyright Act), 110
- Dubois Gérard, 111
- EAI (« Enterprise Application Integration »), 103
- EISA, 96
- ENIAC, 9, 33, 104
- Epson, 114
- Ericsson, 44
- Escarpit Robert, 19
- Ethernet, 6, 9, 90, 93, 95, 96, 98, 112
- Eutelis, 112
- eWorld, 97
- Excel, 96, 103, 118, 119
- Faggin Federico, 33
- Fairchild Semiconductor, 32
- Fielding Roy, 109
- Figer Jean-Paul, 68
- Focus, 103
- Fortran, 102, 111
- Foucault Michel, 25
- Frankston Bob, 117
- Frontline, 119
- FrontPage, 112
- Fylstra Daniel, 117
- Gödel (Théorème de), 67
- Gödel Kurt, 67
- Gates Bill, 33, 78, 87, 89–92, 105–108, 122
- Gem, 115
- Gensollen Michel, 110
- Gerstner Lou, 96
- Gille Bertrand, 30
- Gnumeric, 119
- Groupware, 112
- Grove Andrew, 104
- Grove Andy, 33
- Handspring, 12
- Henry J. S., 112
- Herbsleb James, 109
- Hiltzik Michael, 90, 103
- Hodges Andrew, 74
- Hoerni Jean, 32
- Hoff Ted, 33
- Honeywell, 89
- HP, 96
- IBM, 10, 14, 28, 34, 54, 55, 79, 80, 85, 90, 93–99, 102, 104–106, 113–116, 118, 119
- iMac, 98
- Ingals Dan, 93
- INRA, 86
- INSEE, 38
- Intel, 32–36, 44, 83, 85, 86, 88, 94–97, 99, 104, 105
- Internet, 6, 34, 42, 59, 61, 79, 84, 86, 90, 97, 98, 100, 101, 105, 107, 108, 110, 122
- Intranet, 6, 57, 59, 61, 84, 97, 98, 112, 122
- ISO (International Standard Organization), 25
- Jancovici Jean-Marc, 12
- Java, 93, 103, 108
- Jobs Steven, 90
- Kahn Philippe, 88
- Kapor Mitch, 118
- Kay Alan, 34, 93, 103, 108
- Kenbak-1, 86
- Kidder Tracy, 121
- Kilby Jack, 32
- Knuth Donald, 16, 27, 28, 54, 65, 68, 83
- Kodak, 96
- KSpread, 119
- Lévy Pierre, 68
- Lampson Butler, 93
- Langage, 15
- Langage conceptuel, 17
- Langage connoté, 17
- Laurès Antoine, 112
- Le Diberder Alain, 111
- Le Guyader Hervé, 26
- Lecointre Guillaume, 26
- Lerner Josh, 108
- Levy Steven, 15, 87
- Linux, 103, 119
- Lisa, 33, 93, 95, 98
- LISP, 102

- Lotus, 95, 97, 98, 114, 118–120
Lotus 1-2-3, 95, 115, 118, 119
Lotus Notes, 96, 112
- Médiamétrie, 12
Macintosh, 33, 93, 112, 118, 119, 122
MacOS, 98
Mark-8, 88
Marlowe Jos, 103
Mattessich Richard, 117
McCarthy John, 102
MCI, 96
McLuhan Marshall, 24
McNealy Scott, 103
Merise, 61
Metcalf Robert, 93
Micral, 33
Micropro, 113, 114
Microsoft, 89, 90, 93–100, 104, 105, 107–109, 114–119, 122
Microsoft Works, 118
Middleware, 103
MIPS, 34
MIT, 102, 117
MITS, 88, 89, 91, 92
Mockus Audris, 109
Moore (Loi de), 32, 34
Moore Gordon, 32–34
Morand Bernard, 61
MOS Technologies, 33
Mosaic, 118
Motorola, 33, 36, 96, 99
Mott Tim, 93
MP3, 34
MS-DOS, 113–116
Multiplan, 96, 111, 112, 118
- Nadeau Tom, 109
Netscape, 97
Northern, 44
Notebook, 96, 97
Novell, 96, 100, 105, 116
Noyce Robert, 33
Numérique, 22
- O’Beirne Patrick, 120
Objet, 18
OEM (« Original Equipment Manufacturer »), 100
- OpenOffice, 117
OS/2, 96, 116
OSI (Open Systems Interconnection), 25
Outsourcing, 96, 97
- Palmtop, 11
Panko Ray, 120
Paperback, 118
PARC (Palo Alto Research Center), 85, 90, 93, 95, 103
Pascal, 102, 103
Pattis Richard, 15
Paulson Linda, 109
Pavé Francis, 24
PC, 94
Penny Philippe, 112
Pentium, 34, 96, 97
Pentium Pro, 97
Perl, 103, 108
Perret Jacques, 14
PET, 94
Peterson Pete, 113
Platon, 75
Plotin, 75
Popper Karl, 25, 70, 74
Powerbook, 96
PowerPC, 34
Powers James, 54
Presentation Manager, 116
Prodigy, 97
Proust Marcel, 23
PS/2, 96
- Quattro, 118
Queuille Henri, 81
- R2E, 33, 86
Rational Rose, 103
Raymond Eric, 108
RISC, 98
Roberts Ed, 88
Rock Art, 33
Rosen Benjamin, 118
Russell Bertrand, 67
- Sammet Jean, 101
Scelbi, 88
Scheme, 102, 103
Schneier Bruce, 77
Sears et Roebuck, 101

- Semi-conducteur, 30
- Sendmail, 109
- Serres Michel, 68
- Shannon Claude, 19
- Silicium, 30
- Silicon Graphics, 34, 93
- Simonyi Charles, 93
- Simula, 103
- Sipser Michael, 66, 68
- Smalltalk, 93, 103
- Software Arts, 117, 118
- Solow (Paradoxe de), 96
- Solow Robert, 96
- SPARC, 34
- Sperry Rand, 54
- SQL, 103
- SSEC, 104
- SSI, 113–115
- Star, 93
- Starkweather Gary, 93
- Stiller Richard, 103
- Stroustrup Bjarne, 103
- Sun, 34, 103
- SuperCalc, 118
- Syme S., 110
- Système d'exploitation, 103
- Tableur, 117
- Tandy, 114
- Tanenbaum Andrew, 25
- TCP/IP, 122
- Telser Larry, 93
- Texas Instruments, 34
- Textor, 111
- Thacker Chuck, 93
- ThinkPad, 96
- Tirole Jean, 108
- Titus Jonathan, 88
- TopView, 115
- Transistor, 30
- Treo, 12
- Truong André, 86
- Truong Jean-Michel, 68
- Turing (machine de), 68
- Turing (pari de), 71, 75
- Turing Alan, 68, 70–75
- Univac, 104
- Unix, 96, 100, 103
- VAX, 115
- Victor, 114
- Virilio Paul, 22, 68
- VisiCalc, 117–119
- Visicalc, 90
- VisiCorp, 117, 118
- von Neumann (architecture de), 54, 104
- von Neumann John, 9, 54, 70
- Wal-Mart, 101
- Wall Larry, 103
- WAN (« Wide Area Network »), 96
- Wang, 104, 113, 114
- Watson Thomas J., 54
- Wearable Computer, 11, 123
- Web, 122
- Whitehead Alfred, 67
- Wilder Billy, 13
- Windows, 98, 103, 112, 115, 116, 118, 122
- Windows 1.0, 95
- Windows 95, 97
- Windows NT, 96, 97
- Word, 103, 114, 116, 117
- WordPerfect, 97, 113–117
- Wordstar, 113, 114
- Workflow, 58–60
- Wozniak Steven, 90
- Wysiwyg, 115, 116
- Xénophon, 75
- Xerox, 85, 90, 93, 98, 103
- UML (Unified Modeling Language), 59, 62, 63, 125

Table des matières

I	Comment l'Automate Programmable doué d'Ubiquité assiste l'Être Humain Organisé	3
1	Du côté de l'ordinateur	5
1.1	Évolution du rapport avec la nature	6
1.2	Qu'est-ce qu'un ordinateur?	9
1.3	Diversification de l'ordinateur	11
1.4	Vocabulaire de l'informatique	13
1.4.1	« Ordinateur »	14
1.4.2	« Langage »	15
1.4.3	« Objet »	18
1.4.4	« Donnée » et « Information »	19
1.4.5	« Numérique »	22
1.5	Modèle en couches	25
1.5.1	Couches de l'ordinateur	26
1.5.2	Portée du modèle en couches	29
1.6	Vers l'ordinateur individuel	30
1.6.1	Origines et évolution du microprocesseur	33
1.6.2	La loi de Moore	34
1.7	Évolution du prix des micro-ordinateurs	38
1.8	Conjoncture des TIC	40
1.8.1	Simulation	44
1.8.2	Conséquences	46
2	Automatisme et intelligence	49
2.1	Informatisation de l'entreprise	50
2.1.1	Organisation du travail de bureau: années 1880	51
2.1.2	Arrivée de l'informatique: années 1950	53
2.1.3	La bureautique communicante: à partir des années 1980	57
2.1.4	Du concept au processus: années 1990	58
2.2	Entropie du système d'information	61
2.3	Limites de l'informatique	65
2.4	L'ordinateur et l'intelligence	68
2.4.1	Turing a-t-il perdu son pari?	71
2.4.2	Articuler l'ordinateur et l'être humain	77

2.4.3	Le semi-désordre	80
3	Éclairage historique	83
3.1	Histoire du micro-ordinateur	85
3.2	Les apports du PARC de Xerox	90
3.3	IBM et le PC	94
3.4	Chronologie du PC	95
3.5	Innovations importantes	97
3.5.1	Quelques réussites exemplaires	100
3.6	Les langages de programmation	101
3.7	Le marché de l'informatique	104
3.7.1	Dynamique des équilibres du logiciel	105
3.8	Les logiciels de bureautique	110
3.8.1	Le traitement de texte	113
3.8.2	Histoire du tableur	117
3.9	Esquisse de prospective	122
	Bibliographie	132
	Index	137