

# Un demi-siècle d'Informatique

Gilles Carpentier

Professeur, Responsable du département Informatique à l'ISEP

Je ne vais pas écrire une « brève histoire de l'informatique » pour paraphraser le titre du passionnant ouvrage d'Henri Lilen (« Une brève histoire de l'électronique »), mais plutôt l'évolution de l'informatique telle que je l'ai vécue en tant qu'utilisateur comme tout un chacun, comme acteur dans l'industrie et la recherche et enseignant, notamment à l'ISEP.

## Les années Bull-IBM

Au début des années 60, j'avais entendu parler des ordinateurs, monstres mythiques aux pouvoirs quasi surnaturels, mais sans en constater la réalité jusqu'au jour où l'on m'apportât un listing : « tiens, cela a été fait par un ordinateur » comme s'il s'agissait d'une sainte relique. Le listing consistait en un calendrier avec en son centre La Joconde dessinée avec des croix et des zéros. J'en conclus que les ordinateurs n'étaient pas toujours occupés à des tâches aussi sérieuses que je le supposais. Il s'agissait d'un IBM S/360. Je ne le vis malheureusement jamais, le premier système de cette lignée que j'utilisais plus tard, fut un S/370-158 en cluster.

Mais le premier ordinateur que je pus réellement voir était d'une autre nature, il s'agissait d'un S/3-10 (ou 12D puisqu'il y avait un disque dur). Je fus d'ailleurs plus impressionné par la trieuse de cartes que par l'unité centrale. IBM était à cette époque organisée en deux divisions, la division ordinateurs (DO) pour les mainframes S/370 et la Division des Systèmes de Grande Diffusion (DSGD) pour les minis S/3. Une des raisons invoquées, était la possibilité qu'IBM tombe sous le coup de la loi anti-trust américaine, et qu'il valait mieux que le découpage s'effectue en suivant les pointillés avec deux entités viables indépendamment l'une de l'autre. Il en est résulté deux gammes de systèmes incompatibles jusqu'à l'arrivée de la machine de convergence (l'AS400). Convergence limitée toutefois aux périphériques, mais

la compagnie et ses clients avaient déjà d'autres préoccupations avec l'émergence des systèmes ouverts et l'intrusion de la micro-informatique. Si aujourd'hui IBM se veut le champion des systèmes ouverts avec Linux, que d'efforts, tant techniques que culturels, la compagnie a dû fournir pendant les années 80 et 90. Si je ne prends que l'exemple des réseaux, à peine SNA avait fini sa mue pour être conforme aux normes OSI, que les clients réclamaient l'ouverture à TCP/IP.

Le S/3 fut remplacé dans la seconde moitié des années 70, par un S32 que j'utilisais pour du suivi de projet industriel en construction de machines à papier, à l'aide du logiciel JAS32 (Job Analysis System). Les informaticiens étant tous partis en vacances, je fus bien obligé de me débrouiller avec le langage de commande et quelquefois le GAP II (Générateur Automatique de Programmes ou RPG II en anglais). Il fallait tout de même attendre 11 heures de calcul et autant de temps d'impression pour obtenir un diagramme de GANTT et déterminer le chemin critique.

Au début des années 80, je fus chargé de la migration de 2 IRIS 45 vers un IBM 3 S/38. Je possédais déjà une expérience universitaire de l'IRIS 80. Les systèmes d'exploitation étaient baptisés du même nom (SIRIS) mais n'avaient rien en commun. L'architecture autour de l'IRIS 80 n'était pas banale pour l'époque. Chaque site universitaire possédait un parc de micro-ordinateurs (Commodores 32 ou 64 et Goupil G2) connectés par

RS232 à une passerelle (généralement un CII Mini6) lui même relié par une ligne téléphonique à l'IRIS 80 central. L'étudiant pouvait ainsi s'épargner la perforation des cartes et taper directement son code à l'aide de l'éditeur du Mini6 dans l'émulateur de terminal DKU. Non sans peine toutefois, l'éditeur de l'époque fonctionnait en mode caractère, sans possibilité d'insertion ni de modification, et la suppression n'était possible qu'en tapant un caractère visible à l'écran.

En comparaison, « vi » de Unix était un vrai régal. Pour couronner le tout, le caractère « ! » était à la fois une commande de l'éditeur et un indicateur de début d'instruction pour le langage de commande de SIRIS8. Deux IRIS 45 avec tous les périphériques (lecteurs de disques amovibles, lecteurs de cartes, lecteurs de bandes, perforatrice, imprimantes et télétypes) cela prend de la place et cela occupait à temps plein tout un personnel d'exploitation qui opérait en 3x8 heures, au rythme de la production de l'usine. L'unité centrale était d'une esthétique dont beaucoup ont gardé une nostalgie, façades en aluminium brossé, des voyants à profusion. La visite du service informatique était la vedette des journées portes ouvertes. Il existait déjà sur ce type d'équipement la même angoisse que pour le redémarrage d'un serveur Windows actuel après un changement de version du système : va-t-il redémarrer ? La séquence d'initialisation (on ne disait pas encore « boot »), s'effectuait en entrant directement les instructions dans les registres à l'aide de clés en façade, puis le lecteur de bande prenait le relais pour charger le système et en final le montage des disques. L'opérateur expérimenté n'avait pas besoin de lire sur le télétype, il reconnaissait chaque message au bruit distinctif que l'impression produisait. Un autre son (le fameux « blonck »), signifiait qu'un disjoncteur ou un fusible avait sauté, la caisse à outils était toujours à



IBM 360



CII IRIS 50

portée de la main. La panne la plus fréquente restait le déraillement du fil du contrepoids pour la fermeture des lecteurs de bandes verticaux, la plus sérieuse mais heureusement rare était qu'un disque amovible reste bloqué sur l'axe du lecteur. Mais la panne la plus spectaculaire est la polarisation rémanente d'une mémoire à tores de ferrite. Elle nécessitait le démontage du bloc mémoire (4 kilo octets = 4 kilogrammes) et sa démagnétisation manuelle. Je travaillais alors dans l'industrie automobile, il me suffisait de placer le bloc mémoire sous un électroaimant de deux mètres de diamètre qui servait à soulever les pièces des bacs pour les poser sur le tapis roulant des fours de traitement thermique.

Je suis passé à l'IBM 38 sans l'étape classique de l'IBM 34. L'ancêtre de l'AS400 avec son concept d'espace adressable unique, vision de la mémoire RAM et du disque unifiée, une organisation de toutes les données dans une base unique, un peu Longhorn avant l'heure mais sans l'objet. Son architecture système révolutionnaire pour l'époque était malheureusement souvent mal appréciée des informaticiens chevronnés qui ne retrouvaient pas leurs points de repères habituels (répertoires, fichiers, lecture séquentielle, ...). L'IBM 36 fut alors créé pour rassurer les clients qui voulaient évoluer sans passer à l'IBM 38 tout de suite. Généralement, ils attendirent jusqu'à l'arrivée de l'AS400.

## Avant les PC

Le premier micro-ordinateur d'IBM ne fut pas le PC mais l'IBM 5100. Le 5100 de base ne possédait pas de lecteur de disquette interne et encore moins de disque dur (celui-ci apparût pour le 5120), mais un lecteur de bandes. Sa principale caractéristique était le langage APL, langage symbolique où un caractère représente une instruction complète, d'où un clavier très surchargé et surtout une maintenance difficile. Des programmes très importants ont été mis en œuvre en APL à la fin des années 70, tous n'ont pas été réécrits dans des langages moins exotiques, mais il existe encore une communauté active dans le monde de l'APL, grâce aux professeurs de mathématiques.

Les problèmes liés à la difficulté de la maintenance d'une solution matérielle ou logicielle sont souvent liés à des marchés captifs ou spéciaux. Le Solar16 de la Télémécanique était, à l'origine, conçu pour piloter des processus industriels. Il n'y a que les universités qui essayaient non sans mal (je l'avais planté en installant le compilateur Pascal), de l'utiliser comme un ordinateur plutôt que comme un « process controller ». Mais il fut également utilisé comme calculateur embarqué en militaire, d'où l'obligation de maintenir non seulement un stock de pièces détachées mais aussi des compétences sur cette plateforme. Un autre exemple, est l'environnement logiciel (Prologue, BAL, Pick) d'une application critique (assurances) que l'on continue d'utiliser en émulation sous Unix alors que le matériel d'origine n'existe plus.



Solar 16 de la Télémécanique

### Unix

Je vins à Unix relativement tard. Mais je fis fort, j'achetais un NCR Tower 32 en 1985 sous Unix. Pour pouvoir l'importer directement des États-Unis, je dus remplir les formalités au COCOM (Comité de coordination pour le contrôle multilatéral des échanges Est-Ouest) concernant les restrictions sur les exportations pendant la guerre froide. Et je poursuivis avec un Texas Instruments bi-processeur Intel 286. Le processeur Intel impliquait obligatoirement la version Xenix d'Unix. Xenix est l'Unix de Microsoft, SCO ayant réalisé le portage de Xenix pour processeurs Intel. En fait, j'ai réellement acheté Xenix directement auprès de Microsoft qui l'a distribué jusqu'en 1987. Microsoft a longtemps utilisé Unix en interne, ne serait-ce que pour accéder à l'Internet : il n'y avait pas de serveur DNS sous Windows avant NT4. Un de mes anciens élèves, qui se reconnaîtra peut-être, a été chargé de la maintenance de systèmes Unix chez Microsoft France. Il y a toujours eu des relations entre Microsoft et le monde Unix. Le sous-système Posix au-dessus du noyau Win32 de Windows 2000 et XP est de plus en plus accessible. Les outils de convergence (Interix, SFU) seront intégrés

directement dans Windows 2003 R2. Enfin, mais c'est plus confidentiel, Internet Explorer et Outlook Express pouvaient fonctionner sous Solaris sur Sun Sparc. De son côté, Apple aussi, bien avant MacOS X, a proposé Unix System V (et pas BSD) sur ses matériels sous le nom A/UX. Je m'en suis servi pour créer une passerelle entre AppleTalk/LocalTalk et TCP/IP sur Ethernet.

L'arrivée de la micro-informatique dans les entreprises, n'a pas été vécue de la même manière en Europe et de l'autre côté de l'Atlantique. Aux États-Unis, au début des années 80, la majorité des PME étaient déjà informatisées avec un mini de gestion (style IBM S/34) et des terminaux. Les PC devaient communiquer avec le mini (émulation de terminal, serveur de fichiers, ...) généralement via un réseau local. Tandis qu'en Europe, moins informatisée, une application sous Unix sur un micro-ordinateur pouvait dispenser la PME d'investir dans un mini. Il y eut une floraison de petites SSII déployant ce type de solution dans la première moitié des années 80. Rapidement ces solutions ont été concurrencées par d'autres dès qu'un réseau local de PC sous DOS fut jugé viable, essentiellement grâce à l'émergence de Novell Netware.

### Proteon, Novell Netware

Je participais au déploiement d'un des tous premiers réseaux Novell en France pour un institut de recherche agronomique. Les PC étaient des Olivetti M24 (PC-XT à processeur Intel 8086), le serveur n'avait que 640 Ko de mémoire. Le réseau local provenait de la société Proteon, un anneau à jeton sur câble coaxial, il fallait souder directement les câbles sur le concentrateur pour construire l'anneau. Ni Proteon, ni Novell n'étaient représentés en France, il avait fallu tout importer directement des États-Unis. L'Olivetti M24 était un excellent compatible PC, et ATT les a pendant un temps distribués sous Unix aux USA.

Au début, je rencontrais beaucoup de réticences lorsque je proposais des solutions sur réseaux locaux dans les grandes entreprises. Un article très bien argumenté de Georges Cottin, patron d'Axel, expliquait pourquoi les réseaux de PC étaient une solution inefficace. Il n'avait rien contre le PC, puisqu'il créa l'AXEL 20, un PC compatible IBM avant même la sortie de celui d'IBM. Il fut ensuite commercialisé par Matra sous le nom MAX 20. Fidèle à ses convictions, Axel propose aujourd'hui des terminaux légers avec le minimum de couches embarquées.

La rivalité entre le monde Unix et les réseaux de PC sous DOS se ressentait chez les éditeurs. J'avais trouvé au Texas la société Softcraft, qui proposait une base de données efficace, fonctionnant sous DOS, Netware et Xenix.

Rachetée par Novell, elle abandonna immédiatement la commercialisation de la version Xenix.

## Un peu de Linux, un peu de Microsoft ...

Il est difficile de raconter l'histoire de l'informatique à ce jour, sans mentionner Windows et Linux. Pour Linux, je me contenterai de signaler que le système de fichier ext2 a été écrit par Rémi Card de l'université de Paris VI à l'époque où il était professeur de systèmes d'exploitation à l'ISEP. Mais y aurait-il eu Linux sans le Minix de Andrew Tannenbaum ?

Pour Microsoft, il y a déjà suffisamment de sites retraçant la fulgurante ascension de la société. Je ne mentionnerai qu'un seul produit : Excel, le successeur de Multiplan. De mémoire, j'ai vu fonctionner Excel sous DOS. La première version d'Excel était destinée au MacIntosh. Le copyright d'Excel est 1985, ce qui coïncide avec la sortie d'Excel pour MacIntosh. Ce n'est qu'en 87 qu'Excel fut porté sur Windows 2. Au delà d'Excel, quelle belle invention que le tableur ! Que ferions-nous sans les feuilles de calculs ? Merci à Visicalc.

## La crise du logiciel

Mais les utilisateurs de micro-ordinateurs ne se sont pas contentés de retraiter les données de leur entreprise avec un tableur. En raison de la lourdeur et lenteur de la mise en production d'applications sur mainframes et même minis, ils sont de plus en plus nombreux au début des années 80, à développer leurs propres applications avec des outils que l'on ne qualifie pas encore de RAD (Rapid Application Development). En France, la préférence va à Turbo-Pascal et dBasell. Ce qui posera un double problème : comment intégrer ces applications dans l'architecture du système d'information ? Comment maintenir des applications ne respectant généralement pas les procédures et règles de développement mises en place dans l'entreprise. Lorsque je suis arrivé, jeune ingénieur, à l'informatique de PSA, le premier document qui m'a été remis, était un classeur qui recensait toutes les « normes » d'analyse, de développement (y compris de programmation) et de mise en exploitation d'une application en respectant la méthode CORIG. Les entreprises industrielles n'avaient pas attendu l'avènement d'ITIL (Information Technology Infrastructure Library).

Parallèlement à cette situation, les éditeurs de logiciels tournaient en surchauffe, il fallait occuper ce marché naissant mais en expansion fulgurante qu'est l'informatique. Bref, développer vite, mais pas forcément bien.

Il fallait réagir, et les premières tentatives se focalisèrent sur les langages en apportant les notions de cloisonnement, d'opacité et de modularité avec des langages comme Modula-2 et surtout ADA. Ce qui engendra d'ailleurs un débat (non clos) dans l'enseignement. Faut-il d'abord apprendre un langage bien structuré et rigide comme ADA pour acquérir de bonnes pratiques très tôt et ensuite passer à un langage plus permissif comme le C en gardant les bonnes habitudes ? Ou à l'inverse, commencer par le C qui oblige à comprendre en profondeur ce que l'on écrit afin d'être capable de le déboguer sachant que le seul message d'erreur est le fatidique « SegmentationFault. Core Dumped », avant de passer à un langage de haut niveau, où l'encapsulation et l'abstraction nous éloignent de la réalité du système ?

Mais il fallait une rupture et elle se produisit grâce à l'objet. Pas tant avec C++, car la plupart des programmeurs ne l'utilisaient que comme une amélioration du C, mais avec Java et UML.

Au tournant des années 90, pour mes activités dans le domaine des réseaux locaux, j'avais besoin d'un langage simple, puissant et portable et je trouvai TCL/Tk. Très utilisé encore aujourd'hui chez les équipementiers et les opérateurs de télécommunications, il me permit dans le domaine de la recherche, de commencer à développer des agents mobiles et des moniteurs d'état services en réseau. Développé d'abord à l'Université de Berkeley, son auteur John Ousterhout fut ensuite embauché par les laboratoires Sun. Les similitudes entre la gestion événementielle dans l'interface graphique awt de Java et Tk, ne sont sûrement pas dues au hasard. Pratique pour développer des solutions en RAD, il ne permettait toutefois pas de créer de véritables applications industrielles. Et c'est presque par hasard que je découvris Java. Le jour de sa sortie officielle, je fus invité par Sun à une conférence sur son concept de « Network Computing » ce qui correspondait parfaitement à mon domaine d'activité. Si la conférence s'était appelée « Java », je n'y aurais vraisemblablement pas assisté et j'aurais raté un événement dont les conséquences ont radicalement changé l'informatique. Même si le langage en lui-même souffre de défauts, l'architecture qu'il induit est à la base de la plupart des systèmes d'informations des entreprises d'aujourd'hui.

L'autre facteur structurant, a été l'adoption généralisée d'UML pour la conception objet. L'utilisation du graphisme pour décrire les flux, les procédés, n'est pas nouvelle, mais les ordigrammes étaient restés couchés

sur le papier. Toutefois, en 1991, Borland proposa un outil de RAD graphique nommé ObjectVision. Malgré sa dénomination, cet outil n'était pas orienté objet mais procédural. Le concepteur dessinait un ordinogramme qui était ensuite interprété par un runtime. Malgré une facilité d'apprentissage, il ne connut pas le succès, peut-être un peu trop en avance sur son temps.

### Les loupés

Il est toujours facile de critiquer après coup, les choix stratégiques de certains constructeurs ou éditeurs. Philips qui était un constructeur informatique aurait peut-être obtenu un avantage marketing s'il avait équipé ses PC de lecteurs de CD en standard. Le cas IBM OS/2 est plus complexe. IBM aurait-il dû le développer tout seul (en tout cas, sans Microsoft) ou ne rien faire ? Les ratés ne sont pas l'apanage des sociétés occidentales. Au début milieu des années 80, alors que la tendance générale est à la standardisation, les constructeurs japonais essayaient encore d'exporter des micro-ordinateurs qui n'étaient pas compatibles PC. Ils se rattrapèrent plus tard avec les portables et les imprimantes.

### La standardisation et le progrès

Le problème de la banalisation de l'offre informatique existe et existera toujours. D'un côté, les progrès techniques élargissent le panorama des solutions, mais la loi du marché élimine les plus faibles. Comment se démarquer de la concurrence tout en s'assurant de ne pas s'enfermer dans une architecture propriétaire ? Si l'on regarde le paysage informatique aujourd'hui, on constate que la plupart des constructeurs ont fait le choix de processeurs Intel ou compatibles et d'un système Windows ou Linux. Avec deux exceptions notables : Apple et Sun. Pour Apple, la stratégie est claire, il vaut mieux être le meilleur sur un marché de niche pérenne que d'entrer en compétition avec la meute sur le marché global. Pour Sun, les virages ont été pris trop tard : processeur Sparc obsolète, Solaris maintenu à bouts de bras, même son marché traditionnel en France des universités lui échappe. Et ce n'est pas Java qui va les sauver. Sun n'a pas réussi

à trouver un modèle économique autour de Java alors qu'IBM et d'autres arrivent à créer un business autour de cette plate-forme.

### L'Internet

L'Internet a considérablement modifié la vie des hommes et des entreprises. Personnellement, je pense que l'informatique a plus contribué au développement et à l'évolution de l'Internet que l'inverse. Je préfère parler de co-évolution, tout progrès du réseau (débit, mobilité, ...) permet d'envisager de nouveaux usages ; tout progrès de l'informatique, ajoute de nouveaux services accessibles par le réseau et contribue à la croissance de son utilisation.

Le modèle client-serveur, le web et l'HTML sont des concepts informatiques qui ont trouvé dans l'Internet un vecteur pour leur généralisation. Par contre, les web services n'ont d'intérêt que dans le cadre de l'Internet. L'Internet sur radio, les réseaux ambiants créent de nouvelles possibilités que l'informatique exploitera en offrant de nouveaux services.

### Vers la disparition de l'objet ordinateur ?

Au début de la révolution industrielle, chaque atelier possédait un unique moteur universel d'où partaient des courroies pour entraîner les différentes machines. Puis le moteur s'est banalisé, miniaturisé et surtout spécialisé. Il est devenu une composante de la machine, à tel point qu'il a disparu, puisqu'il est partout.

Lorsque l'ordinateur sera partout, il disparaîtra en tant qu'objet en soi. Mon vieux four à micro-ondes possède 2 microprocesseurs du même type que celui de mon (encore plus vieux) Commodore 64. Je n'utilise pas mon four pour mettre à jour mon CV, je n'utilise pas mon lecteur de DVD pour cuire une pizza.

La spécialisation par fonction commence déjà à se ressentir avec les téléphones portables. Téléphoner, c'est communiquer, lire et envoyer ses mails. En Corée, plus de la moitié des mails sont envoyés depuis des téléphones portables.

#### Gilles Carpentier

Professeur et responsable du département Informatique à l'ISEP. Il a essentiellement exercé sa carrière en Informatique dans différents secteurs comme l'industrie automobile, le papier carton, l'édition, le domaine pharmaceutique et les télécommunications.