

1 L'évolution de l'informatique

Depuis la création de l'ordinateur, la taille des machines a beaucoup évolué et l'utilisation de l'informatique également. Actuellement, l'informatique a envahi tous les domaines de la vie courante, et il serait plus facile de dresser la liste des domaines où elle n'intervient pas, que celle où elle intervient. Mais, comment en sommes nous arrivés aux machines d'aujourd'hui ? Commençons par un peu d'histoire.

1.1 Historique

Les inventions et découvertes qui nous ont conduits aux ordinateurs tels que nous les connaissons aujourd'hui sont nombreuses. Aussi, les quelques lignes qui vont suivre n'ont pas pour prétention d'être exhaustives, tout au plus, nous présentons quelques points de repaires, entre lesquels il y a bien des omissions. Les découvertes peuvent être classées en deux catégories : celles ayant trait au raisonnement, et celles concernant la partie physique. Encore que ces deux catégories soient souvent intimement liées, tels l'algèbre de Boole et les circuits électroniques.

De nos jours, il n'est un secret pour personne que, dans l'ordinateur, les informations sont codées sous forme binaire. On retrouve de traces de l'utilisation de la notation binaire 3000 ans avant notre Ere, sous l'empereur Chinois Fou-Hi dont le symbole magique, l'octogone à trigramme contient les huit premiers nombres entiers représentés sous forme binaire par des traits interrompus ou non : 000 001 010 011... Ce ne sera qu'en 1679 que Leibnitz découvre et analyse les octogrammes de Fou-Hi, et met au point une arithmétique binaire.

Pendant ce temps, vers l'an -300, le grec Aristote définit, dans ses écrits, les bases de la logique. En 1854 l'anglais Boole publie un ouvrage dans lequel il démontre que tout processus logique peut être décomposé en une suite d'opérations logiques (ET, OU, NON) appliquées sur deux états (ZERO-UN, OUI-NON, VRAI-FAUX, OUVERT-FERME). Puis en 1938, dans sa thèse, Shannon est le premier à faire le parallèle entre les circuits électriques et l'algèbre booléenne. Il définit le chiffre binaire : bit (BInary digiT).

Les premiers instruments de calcul apparaissent au moyen orient vers -500, avec les bouliers et les abaquas. Bien des expériences ont été faites ensuite, mais il faut attendre le XVIIIème siècle pour que des machines à calculer fonctionnant réellement voient le jour : en 1624 l'Allemand Schickard présenta sa machine mais elle fut détruite la même année. En 1642, la Pascaline du français Blaise Pascal fut la première à réellement fonctionner. Elle était capable de faire des additions et des soustractions. Beaucoup de personnes s'appuieront sur cette découverte. Ainsi, Leibnitz construisit sa propre machine en 1694 (divisions et multiplications) en suivant le modèle de Pascal. Quelques années plus tard, en 1833, l'Anglais Babbage crée la machine analytique. Cette machine ne put jamais fonctionner, car elle était entièrement mécanique et la technologie de l'époque ne permettait pas de fabriquer des pièces avec une précision suffisante. Cette machine intégrait beaucoup de concepts de l'ordinateur contemporain : mémoire, registre, unité de calcul.

Les machines à calcul ne sont pas à dissocier des méthodes de calcul. Les égyptiens inventèrent la trigonométrie pour recalculer les propriétés de chacun après le retrait du Nil, suite aux inondations. Le calcul du mouvement des planètes et des astres s'est fait, au début, sans machine. En 820, le mathématicien al'Khwarizmi publie à Bagdad un traité intitulé "La science de l'élimination et de la réduction" qui, importé en Europe Occidentale lors

des invasions Arabes, aura une grande influence sur le développement des mathématiques. En 1840, collaboratrice de Babbage, la mathématicienne Ada Byron, comtesse de Lovelace, définit le principe des itérations successives dans l'exécution d'une opération. Elle créa les premiers programmes afin de les utiliser sur la machine analytique de Babbage. En l'honneur du mathématicien Arabe al'Khwarizmi, elle nomme le processus logique d'exécution d'un programme : *algorithme*. Elle est la mère de l'algorithmique moderne, cent ans avant que le premier ordinateur ne fonctionne vraiment. Ada est morte très jeune, à l'âge de 37 ans.

Pour la partie communication, outre l'utilisation de codes sonores (tamtam,...) ou visuels (message avec de la fumée,...), en 1838, le peintre Américain Samuel Morse invente le code qui porte son nom utilisant des points et des traits pour représenter les caractères à transmettre. Entre 1836 et 1838, les Anglais Edward Davy, William Looke et Charles Wheatstone vont inventer et mettre au point le télégraphe. Le premier télégraphe fonctionnera en 1844. Le premier câble transatlantique fut tiré entre les Etats Unis et l'Europe pour interconnecter les systèmes de communication Américains et Européens. Il cessa de fonctionner au bout de quelques jours ! Un second câble transatlantique fut tiré en 1866 et resta en exploitation pendant une centaine d'années. En 1876, l'Américain Graham Bell invente le téléphone et fonde la compagnie Bell Telephone Company.

Enfin, l'ordinateur pu être créé grâce aux progrès sur les circuits électriques. En 1919 Eccles et Jordan inventent le basculeur à partir de deux triodes. Plus connu maintenant sous le nom de flip-flop ou circuit bi-stable. Quelques années plus tard, en 1937 : George Stibitz crée le premier circuit binaire, un additionneur. Il l'appelle le Model K (pour Kitchen) car il l'a créé dans sa cuisine à partir d'une planche à pain !

A présent tout est prêt pour concevoir le premier ordinateur. En 1937, l'anglais Alan M. Turing publie un document sur les nombres calculables. Il résolvait des problèmes mathématiques en utilisant une sorte d'ordinateur logique très simple appelé depuis Machine de Turing : une bande de papier comportant des cases, des pions à mettre sur ces cases, un trombone pointant sur la case courante et un tableau d'instructions conditionnelles à réaliser. En 1940, Pour décrypter les messages de l'armée Allemande, les Anglais mettent au point sur le site de Bletchley Park les calculateurs Robinson et Colossus sous la direction du mathématicien Alan Turing. Ce sont les premières machines qui intègrent les concepts d'arithmétique binaire, d'horloge interne, de mémoire tampon, de lecteurs de bande, d'opérateurs booléens, de sous-programmes et d'imprimantes. Tout ceci restera inconnu car "Secret défense" jusqu'en 1975. Le Colossus voit le jour en 1943, ainsi que le premier langage de programmation binaire inventé par Alan Turing.

1943 est aussi l'année où John von Neumann publie son modèle de traitement de l'information. C'est le concept de base sur lequel sont bâtis les ordinateurs. C'est un concept mathématique et algorithmique, un automate. Cette même année, voit la création du ASCC Mark I (Automatic Sequence-Controlled Calculator Mark I) à Harvard par Howard Aiken et son équipe (avec le soutien d'IBM). C'est un énorme calculateur électromécanique (3000 relais, 800 km de câbles) qui permet de faire 3 opérations sur 23 chiffres par seconde. Cette machine était très proche dans son principe de fonctionnement des plans de la machine analytique de Babbage. Le programme était lu depuis une bande de papier. Les données à traiter pouvaient être lues depuis une autre bande de papier ou un lecteur de cartes. Les branchements conditionnels n'étaient pas possibles. Par la suite, la machine fût modifiée pour permettre, grâce à d'autres lecteurs de bandes, d'aller lire une séquence d'instructions sur un autre lecteur, permettant ainsi les branchements conditionnels ou le lancement de sous programmes. Elle est considérée comme le premier ordinateur, en l'absence de la connaissance du Colossus.

Comme indiqué un peu plus loin, sur ce type de machine, chaque programmeur fabriquait son programme sur un circuit qu'il venait enficher sur l'ordinateur pour le faire fonctionner. En 1945, un insecte coincé dans les circuits bloque le fonctionnement du calculateur Mark I. La mathématicienne Grace Murray Hopper décide alors que tout ce qui arrête le bon fonctionnement d'un programme s'appellera BUG. Cette paternité est toutefois controversée car que le terme BUG était déjà utilisé bien avant cela : Thomas Edison par exemple, (seconde moitié du 19e siècle), avait employé ce terme dans un courrier où il parlait de la mise au point problématique de l'une de ses inventions.

Les choses vont ensuite aller très vite. En 1950 : Invention de l'assembleur par Maurice V. Wilkes de l'Université de Cambridge. Auparavant, la programmation s'effectuait directement en binaire. Un an plus tard, Grace Murray Hopper invente le premier compilateur. Il permet de générer un programme binaire à partir d'un code source. L'invention du transistor (1947) puis des circuits intégrés (1958) vont permettre de rendre les ordinateurs plus efficaces après leur intégration dans les machines en 1955 et 1968 respectivement. 1952 voit la mise en place du réseau militaire SAGE. Le premier réseau à but commercial est créé en 1955. Il s'agit d'un système de réservation de places sur des avions aux USA reliant 1200 télécopieurs. 1969, création du réseau ARPANET, ancêtre d'internet. En 1971, 23 ordinateurs sont connectés sur ce réseau, sur 15 sites différents, à 50 kbits par seconde. 1976, création de TCP/IP. Internet est né en 1980 de l'interconnection des réseaux ARPANET et CSNET. 1989 voit l'explosion des réseaux de communication.

Côté langages, permettant une programmation plus aisées des ordinateurs, en 1957, création du FORTRAN par John Backus. En 1958, John Mac Carthy crée le langage LISP, premier langage d'intelligence artificielle. En 1970, Ken Thompson, pensant que le système d'exploitation UNIX ne serait pas complet sans un langage de programmation de haut niveau crée le langage B. 1971-1973 : Dennis Ritchie du Bell Lab d'ATT reprend le langage B écrit par Ken Thompson le fait évoluer et le dote d'un vrai compilateur générant du code machine (le B était un langage interprété). Ainsi né le langage C. Ce langage est à la fois proche du matériel, permettant ainsi de réécrire le noyau UNIX en C et suffisamment généraliste, le rendant ainsi facilement portable. Les développements et les succès du langage C et d'UNIX sont intimement liés. 1971 voit la création du langage de programmation logique Prolog par Alain Colmerauer (Marseille) et Philippe Roussel, et 1972 celle du premier langage orienté objet, SmallTalk par Alan Kay au Xerox PARC. 1975 apparition du Basic et de SQL. En 1983, Bjarn Stroustrup développe une extension orientée objet au langage C : le C++. En 1991, un groupe de travail de Sun crée le langage Oak rebaptisé Java en 1995.

1.2 Les systèmes d'exploitation

Les logiciels se répartissent en deux grandes catégories : les programmes systèmes qui permettent le fonctionnement de l'ordinateur, et les programmes d'application qui résolvent les problèmes des utilisateurs. Le système d'exploitation est le plus important des programmes systèmes. Il contrôle les ressources de l'ordinateur et fournit la base sur laquelle seront construits les programmes d'application.

Le but du système d'exploitation est de libérer les programmeurs de la complexité du matériel. Il consiste à enrober le matériel avec une couche de logiciel qui gère l'ensemble du système. En fait, le système d'exploitation présente au programmeur une interface ou machine virtuelle plus facile à comprendre et à programmer. Le système d'exploitation a également une autre fonction, celle de gestionnaire des ressources de l'ordinateur : connaître à tout moment l'utilisateur d'une ressource, gérer les accès à cette ressource, en accorder

l'usage et éviter les conflits d'accès entre différents programmes ou utilisateurs.

La programmation d'un ordinateur est organisée en couches. En dessous du système d'exploitation se trouve les couches matérielles avec, au niveau le plus bas les dispositifs physiques (circuits intégrés, fils électriques, alimentation, disques,...). Elle concerne l'ingénieur en électronique. Vient ensuite un logiciel de base qui contrôle ces différents dispositifs et fournit une interface plus simple à la couche suivante. Ce logiciel est appelé un microprogramme . Il se situe dans une partie de la mémoire qui ne peut être modifiée : la ROM (Read Only Memory). C'est un interpréteur des commandes de la couche suivante : le langage machine (assembleur).

Outre le système d'exploitation, le système est également constitué de différents logiciels qui permettent d'aider l'utilisateur dans les différentes tâches qu'il doit accomplir. Des fonctions de recherche de fichiers et de répertoires, de fusion de fichiers, de création de répertoire. Un interpréteur de commandes (le shell), des compilateurs, des éditeurs, et bien d'autres utilitaires de configuration. Tous ces programmes ne dépendent pas des programmes d'application, mais sont là pour aider à les développer et à les utiliser. Contrairement au système d'exploitation qui est le seul à s'exécuter en mode noyau (dit encore mode superviseur), ces autres programmes d'aide s'exécutent en mode utilisateur, c'est-à-dire, à la demande de l'utilisateur. Il existe d'ailleurs plusieurs logiciels permettant de réaliser une même tâche d'aide à l'utilisateur, et ce dernier à le choix du logiciel qu'il peut utiliser : quel compilateur ? quel éditeur ? Une confusion fréquente fait appeler système d'exploitation l'ensemble du système : système d'exploitation et utilitaires (Linux, Windows,). Le système d'exploitation ne dépend que du constructeur. Voici comment ont évolués les systèmes d'exploitation au cours du temps :

Premier ordinateur numérique : Charles Babbage (1792-1871)

Echec de son fonctionnement car entièrement mécanique et la technologie de l'époque ne permettait pas de construire des pièces à la précision requise.

Première génération d'ordinateur (1945-1955)

Les machines énormes sont construites au moyen de tubes électroniques. Un seul groupe de personnes concevait, construisait, programmait, utilisait et effectuait la maintenance d'une machine. Une tranche de temps était réservée à chaque programmeur qui enfichait son circuit (ou son lot de cartes perforées à partir de 1950) pour faire fonctionner l'ordinateur. Tous les programmes réalisaient des calculs numériques.

Deuxième génération (1955-1965) : Avènement des transistors et du traitement par lots

Les ordinateurs sont plus fiables et commencent à être commercialisés (grandes sociétés, administrations, universités). Séparation nette entre concepteurs, constructeurs, opérateurs, programmeurs et personnel de maintenance. Pour exécuter un programme, le programmeur devait écrire le programme, le donner à perforer sur des cartes, le transmettre à un opérateur, qui le plaçait dans le lecteur de cartes dès qu'il y avait un créneau de libre, chargeait au besoin le compilateur nécessaire. Après exécution, il récupérait le listing des résultats. Ceci prenait beaucoup de temps en déplacement et en transition. Pour limiter des pertes de temps, le

traitement par lots fut créé. Il consiste à faire lire un ensemble important de travaux sur une première machine peu puissante et à les enregistrer sur une bande magnétique. Ensuite cette bande est transférée sur une machine plus puissante qui les lit et les traite l'un après l'autre, enregistrant les résultats sur une bande magnétique. Une fois tous les travaux de ce lot traités, la bande est transférée sur une machine moins puissante du même type que la première pour imprimer les résultats. (Premiers systèmes d'exploitation, dont les cartes de contrôle sont les précurseurs des langages de contrôle).

Troisième génération (1965-1980) : Les circuits intégrés et la multiprogrammation

Cette période marque la disparition des gammes de machines spécifiques (calcul scientifique ou bien traitement caractères pour les banques et les entreprises). Une même machine plus ou moins puissante, pouvait traiter tout type de travaux. Il en résulte, avec l'avènement des circuits intégrés, un meilleur rapport coût/performance, mais des systèmes d'exploitation énormes, remplis de bogues qu'il fallait sans cesse corriger.

Cette période a vu l'apparition de la *multiprogrammation* : Cette technique a permis de réduire considérablement les temps d'attente des processeurs lors des opérations d'entrée/sortie (80 à 90% du temps total), surtout sur les applications commerciales. Elle consiste à partitionner la mémoire. Chaque partie contient un travail à exécuter. Lorsqu'un travail en cours d'exécution doit attendre la fin d'une opération d'entrée/sortie, un autre travail peut utiliser le processeur qui, sans cela, serait inactif. Cette présence simultanée de plusieurs travaux en mémoire nécessite des circuits de contrôle pour protéger chaque travail contre les intrusions et les erreurs des autres.

Autre technique : la *spoule* (En anglais SPOOL Simultaneous Peripheral Operation On Line). La spoule est utilisée en entrée et en sortie et consiste à stocker les travaux en attente. Par la suite, dès qu'une partition se libère (fin d'un travail), un travail en attente est immédiatement chargé dans celle-ci, et son exécution lancée. Actuellement, tous les ordinateurs utilisent cette méthode, ainsi que le partage de temps.

Le *partage de temps* (*Time sharing*) : C'est une variante de la multiprogrammation dans laquelle chaque utilisateur possède un terminal en ligne. Dans un système à temps partagé, si 20 utilisateurs sont simultanément connectés, et si 17 d'entre eux réfléchissent, discutent ou boivent le café, le processeur est réparti entre les trois tâches restantes. Souvent, les tâches réalisées sont courtes. L'ordinateur peut ainsi répondre rapidement à un nombre important d'utilisateurs et peut même continuer un traitement par lots en arrière-plan lorsque le processeur est libre. (Création du système d'exploitation UNIX).

Quatrième génération (1980-1990) : Les ordinateurs personnels

L'époque de l'ordinateur personnel est venue avec le développement de *circuits LSI* (Large Scale Intergration), qui contiennent des centaines de transistors au centimètre carré. Large disponibilité de puissance de calcul, développement de l'interactivité et du graphisme, ont entraîné une importante industrie de production de logiciel pour les ordinateurs individuel. Une grande partie de ces logiciels sont conviviaux, c'est-à-dire destinés à des utilisateurs peu compétents en informatique, et qui ne désirent, absolument pas s'y initier. Le développement de ces logiciels et celui des réseaux ont permis à l'informatique d'envahir peu à peu tous les domaines de la vie courante.

Deux systèmes d'exploitation dominent le marché des ordinateurs personnels et des stations de travail : *MS-DOS* suivi de Windows, écrit par Microsoft, et *Unix*. Actuellement, il existe aussi des systèmes d'exploitation distribués. Dans ces systèmes, les utilisateurs n'ont pas à savoir où se trouvent leurs fichiers et où leurs programmes sont exécutés. Cette tâche incombe au système d'exploitation qui doit s'en acquitter de manière transparente pour l'utilisateur. Ces systèmes permettent l'exécution simultanée des programmes sur plusieurs processeurs (voire plusieurs machines). Les algorithmes d'ordonnancement sont plus complexe afin d'optimiser le parallélisme.

1.3 Structure d'un ordinateur

Très schématiquement, un ordinateur est un système informatique qui se veut dynamique. Par suite,

1. Il est le siège d'une activité qui porte sur

La conservation de l'information : c'est la mémorisation. Cette fonction est assurée par la *mémoire centrale*, dite RAM (Random Access Memory).

La création : c'est le traitement de l'information. Fonction assurée par un ou plusieurs *micro-processeurs*.

Le transfert d'information : Communication entre les différents éléments. Fonction assurée par les *bus*.

2. Il effectue des échanges avec son environnement. Fonction assurée par un ou plusieurs *boîtiers d'entrée/sortie*. Ces échanges se font avec :

- des opérateurs humains : clavier, console, souris, imprimante,...
- des processus physiques : supports de disques durs, disques souples, mémoire flash, disques compacts, bandes magnétiques, robots,...
- d'autres systèmes informatiques : réseaux, modems,...

Les échanges sont définis par des suites d'instructions préalablement enregistrées dans une toute petite partie de la mémoire centrale, inaccessible à l'utilisateur, appelée R.O.M. (*Read Only Memory*), que rien ne peut modifier, et qui sert principalement au démarrage de l'ordinateur, et surtout par le système d'exploitation chargé de la gestion de l'ordinateur. L'autre partie de la mémoire centrale, la plus importante, appelée R.A.M. (*Random Access Memory*), est accessible à l'utilisateur. Lorsqu'on éteint l'ordinateur, les informations contenues dans la R.A.M. sont perdues, alors que celles se trouvant dans la R.O.M. demeurent. D'où les termes parfois employés de mémoires *permanentes* et mémoires *temporaires* ou mémoire *vive*.

Un micro-processeur est un composant capable d'interpréter et d'enchaîner l'exécution d'une séquence d'*opérations élémentaires*, selon un codage défini par le constructeur. Un *programme* binaire, exécutable, est une séquence d'instructions élémentaires.

La gestion de la mémoire centrale et du boîtier d'entrées/sorties est assurée par le *système d'exploitation*.

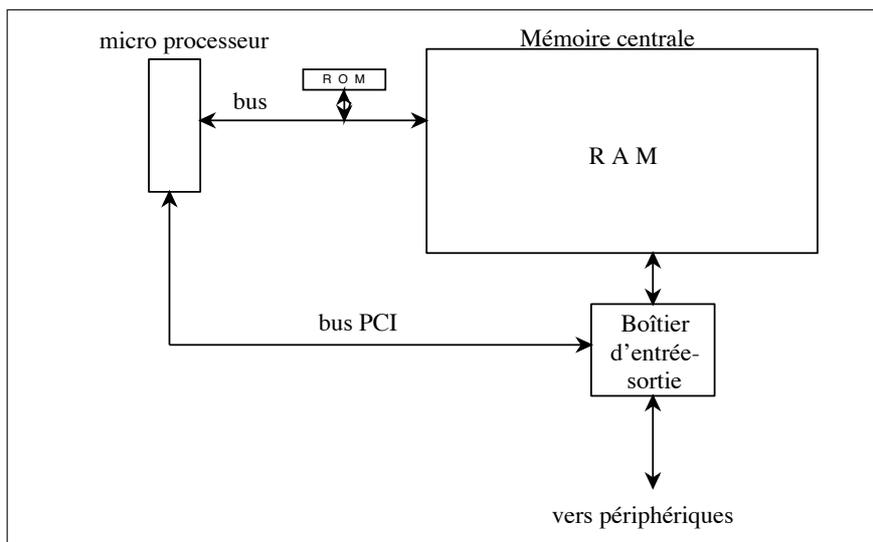


Figure I.1: vue schématique d'un ordinateur

1.4 Les couches d'un ordinateur

Un ordinateur et ses logiciels représentent une hiérarchie d'abstractions appelées *machines virtuelles*. Comme nous l'avons vu dans la section précédente, ces abstractions appelées aussi *couches*, sont apparues au cours du temps pour faciliter l'utilisation de l'ordinateur et par là même, son accession à de plus en plus de monde, dans de plus en plus de domaines. Les premières couches sont très proches de la machine et donc réservées à un petit nombre de spécialistes. Les dernières sont très évoluées et permettent pratiquement à *Monsieur tout le monde* de faire de l'informatique. Chaque couche est bâtie sur la couche qui la précède; chacune de ses instructions est un petit programme plus ou moins complexe composé d'instructions de la couche sur laquelle elle repose et destiné à réaliser une action. Ainsi, par un effet multiplicateur, une instruction d'une couche supérieure représente peut être 1000 instructions de la couche la plus basse, voire beaucoup plus. L'utilisateur d'une couche supérieure n'a pas à se soucier du fonctionnement profond de l'ordinateur. Son travail est grandement simplifié. Il va pouvoir consacrer plus de temps à la logique de son système.

Niveau	Abstraction
6	programmes applicatifs
5	langages de programmation
4	langage assembleur
3	noyau du système d'exploitation
2	langage machine
1	micro-programmation
0	logique numérique

0 Abstraction des circuits électroniques.

1 Premier niveau de langage : séquences d'étapes utilisées pour réaliser les instructions du niveau 2.

- 2
- ajouter deux nombres,
 - déplacer des données,

- déterminer si un nombre est plus grand qu'un autre,
- ...

3 Le système d'exploitation.

4 Représentation symbolique des instructions des niveaux inférieurs.

5 Basic, Fortran, C, Pascal, Cobol, C++, Java,...

6 Les applications.

1.5 Les types de langage de haut-niveau

Il existe plusieurs types de langage. Chacun d'eux a sa spécificité. Voici les principaux types et les principaux langages qui les représentent :

classique ou procédural : Dans ce type de langage, les opérations à réaliser sont décrites pas à pas. Le présent ouvrage traite de ce type de langage. Voici quelques uns d'entre eux : Fortran, Pascal, Cobol, C,...

fonctionnel : Ici, c'est l'aspect fonction qui est privilégié. Représentants : Lisp, ML, Scheme,...

logique : La logique est la base de ces langages. La programmation se fait en définissant des règles du type $A \Leftarrow B$. Représentants : Prolog, Programmation Logique avec Contraintes (PLC) : Prolog IV, CHIP, CLP(R), Programmation Concurrente avec Contraintes (CC): AKL,...

objet : Ce type de programmation favorise l'autonomie des objets, leurs responsabilités et l'organisation entre les entités du systèmes. Représentants : Smalltalk, C++, Objective C, Java,...

fonctionnel, logique et objet : Langages hybrides tirant le meilleur parti de chacun des types sous-jacents. Représentant : Oz,...

événementielle, interactive, langages de requêtes : Langages appropriés pour traiter des événements, ou pour attaquer des bases de données, ou pour dialoguer avec un système informatique. Représentant : SQL,...

2 Phases de vie d'un logiciel

La construction d'un logiciel ne se restreint pas à de la programmation. Bien des étapes sont nécessaires avant d'écrire une ligne de code. Ces étapes sont à respecter si l'on veut un travail correct, soigné, fiable, maintenable, évolutif, robuste. La partie codage représente souvent un temps minime par rapport aux autres phases de la construction. On peut résumer le processus d'élaboration par les étapes suivantes :

- a) La demande
- b) L'analyse de la demande
- c) La conception