

Comprendre les médias interactifs

Louis-Claude Paquin

2 L'automate

2	L'automate	1
2.1.	Cartographie.....	3
2.1.1.	Instrumental ou spéculatif	4
2.1.2.	Étymologie	5
2.1.3.	Machine régulée	6
2.1.4.	Machine à programme.....	7
2.1.5.	Temporalité	10
2.1.6.	Déterministes ou auto-adaptatifs	10
2.1.7.	Un système informatique abstrait	11
2.2.	Approche instrumentale.....	12
2.2.1.	L'horloge	13
2.2.2.	Le calculateur arithmétique	15
2.2.3.	Le calculateur analytique	17
2.2.4.	Calculateurs électromagnétiques ou électroniques ?	19
2.2.5.	L'automate industriel.....	20
2.3.	Approche spéculative	23
2.3.1.	Le miroir	24
2.3.2.	Le fantasme.....	25
2.3.3.	Ésotérisme	27
2.3.4.	Quelques jalons.....	27
2.3.5.	Structure du récit mythique	29
2.3.6.	Masques et statues antiques	29
2.3.7.	Le jaquemart	31
2.3.8.	Le théâtre de machines	31
2.3.9.	Les machines à musique.....	32
2.3.10.	Animaux et androïdes	34
2.4.	Les automates digitaux	36
2.4.1.	Le calcul logique.....	36



Comprendre les médias interactifs de Louis-Claude Paquin est mis à disposition selon les termes de la licence Creative Commons 4.0 : Attribution - Pas d'utilisation commerciale – Pas de modification.

2.4.2.	La machine de Turing	38
2.4.3.	L'ordinateur	40
2.4.4.	Les variables.....	42
2.4.5.	Les listes	43
2.4.6.	Le principe de décomposition	44
2.4.7.	Les fonctions.....	45
2.4.8.	Modularisation.....	47
2.4.9.	L'approche orientée objet : une société d'automates	48
2.5.	L'intelligence artificielle	51
2.5.1.	Connexionnisme versus rationalisme	52
2.5.2.	Test de Turing	53
2.5.3.	L'intelligence en question	54
2.5.4.	Utopie motivante	55
2.5.5.	Les réseaux neuronaux.....	55
2.5.6.	La planification automatique.....	56
2.5.7.	Des systèmes experts	58
2.5.8.	La représentation des connaissances	59
2.5.9.	Cumul de l'incertitude	60
2.5.10.	Limites des systèmes experts	61
2.5.11.	Des agents intelligents.....	62
2.6.	Les automates médiatiques.....	64
2.6.1.	La navigation Web	64
2.6.2.	Les moteurs de recherche.....	65
2.6.3.	Le tableau, mécanisme du puzzle.....	67
2.7.	Bibliographie.....	68



Avec les médias interactifs émerge indiscutablement un nouveau phénomène culturel, social et économique. La plupart du temps, on les oppose aux médias « linéaires », ainsi qu'aux pratiques culturelles qui existent depuis l'aube de l'humanité. Il n'est pas rare de lire, sous la plume de doctes spécialistes, des textes marqués par l'idéologie du progrès, qui présentent les médias interactifs comme une *révolution*. En substance, leurs idées peuvent se reformuler comme suit: en proposant un langage médiatique nouveau, les médias interactifs font émerger des pratiques communicationnelles radicalement nouvelles qui changent notre manière d'appréhender le monde.

Ici, on propose l'approche inverse: notre idée de départ, c'est que pour comprendre l'essence et, accessoirement, la nouveauté d'un certain phénomène, on doit d'abord l'ancrer dans le passé, identifier les jalons de l'histoire qui aboutissent à son émergence, tenter d'établir des filiations, effectuer des dérives, bref, se lancer dans les opérations cognitives qui font l'essence du « comprendre ».

Les médias interactifs s'ancrent dans deux phénomènes, l'automate, objet de discussion de ce chapitre, et le spectacle, abordé dans le chapitre suivant.

Le but de ce chapitre, c'est de montrer qu'il est pertinent, si l'on veut comprendre la mécanicité informationnelle fondée sur le langage des énoncés d'instructions organisées en programmes, d'étudier les connaissances et le savoir-faire développés depuis l'Antiquité autour de *l'automate instrumental et spéculatif*. En effet, l'automate numérique que l'on retrouve dans les médias interactifs s'inscrit dans le même genre de contexte épistémologique que l'automate à moteur.

2.1. Cartographie

L'engouement contemporain pour les automates n'est pas récent. Depuis l'Antiquité, on recense des traités consacrés aux automates et à l'automatisme, qui contiennent tantôt des descriptions détaillées d'automates, tantôt des connaissances techniques, mais aussi ésotériques sur le sujet. Le terme « automate » renvoie bien plus qu'à une classe de machines; il renvoie, en réalité, à un *schème universel*, à savoir celui de l'ambivalence qui se joue entre une *appréhension spéculative* et une



appréhension instrumentale de l'univers. Le thème de l'automate a fasciné la quasi-totalité des cultures et des époques. A ce titre, cette thématique est véritablement transhistorique.

2.1.1. *Instrumental ou spéculatif*

Donc, le terme et partant le concept d'automate existent au moins depuis l'Antiquité. Dès cette époque, le terme présentait une ambiguïté fondamentale. En effet, il s'inscrit dans deux domaines de référence distincts, voire diamétralement opposés. Premièrement, il appartient au *domaine de l'artisanat*, et y désigne une pratique qui, systématisée, se transformera en une technique, dont une certaine partie, plus abstraite, constituera à son tour une science à part entière. Deuxièmement, il appartient au *domaine de l'imagination et de la création*. Cette ambiguïté est fondamentale, car elle se retrouve au plan de la finalité de l'automate. En effet, d'un côté, il y a les *automates instrumentaux*, c'est-à-dire des machines à programme conçues pour accomplir une tâche donnée dans un environnement donné. D'un autre côté, il y a les *automates spéculatifs*, purs produits de l'imagination, et non de la raison. Ces créations génèrent à leur contact un *effet miroir*. En particulier, elles nous renvoient, surtout lorsqu'ils se font androïdes, à notre image d'humain.

Une autre dualité vient croiser cette dichotomie instrumental – spéculatif. Cette fois, celle-ci touche plutôt à l'*ontologie* des automates: correspondent-ils à des artefacts matériels, ou à des constructions immatérielles, abstraites ?

Ont fabriqué des automates des artisans maîtrisant la mécanique au même titre que les maîtres horlogers. Les grandes cathédrales étaient dotées d'horloges monumentales auxquelles étaient adjoints des *jaquemarts* (définis plus bas); on doit entre autres à Vaucanson un canard et un flûtiste (appelé alors *Flûteur*). Ces automates reposent sur un mécanisme composé d'engrenages, de ressorts, de leviers, etc. A Souillac, en France, un musée est même consacré aux automates.

D'autres constructions ne s'incarnent pas matériellement, et restent à l'état de projet ou de description, soit parce que leur réalisation dépasse les capacités



humaines, comme le Golem, soit parce qu'elles s'expriment dans un langage d'instructions réalisées par un ordinateur. L'interactivité du multimédia est assurée par des automates. Ceux-ci, bien que de nature différente, présentent certaines similitudes avec les autres, à cause de la filiation de l'ordinateur avec le monde des machines (cf. 1.5.2).

Entre ces deux pôles de l'instrumental et du spéculatif, la pensée mécaniste qui débouchera sur l'automate se trouve déjà présente dans la philosophie ancienne, notamment dans la conception de l'homme. L'automate joue un rôle heuristique en fournissant certaines catégories d'appréhension et d'analyse du monde. Autour de l'automate se sont développés des connaissances, une pensée et un savoir-faire à l'origine de la révolution industrielle du 19^e siècle, et de la révolution informatique du 20^e. Les automates des machines analogiques se sont alors converties en machines digitales et immatérielles.

2.1.2. Étymologie

Le terme *automate* se retrouve déjà dans les dictionnaires de grec ancien. Il est décomposable en deux parties : le préfixe *auto-* et la racine lexicale *-matée*. Le préfixe lui-même n'est pas ambigu. Il établit une référence circulaire à l'identité de l'individu, et se traduit, suivant le contexte, par « le même », « lui-même » ou « de lui-même ». Le radical, *lui*, est ambigu. Il est tantôt rattaché à la racine du verbe *menomenai*, qui, appliqué à des sujets humains, exprime la force et la tension psychologique, la passion, le désir ou l'intention. Mais il est aussi rattaché au radical indoeuropéen *men*, qui correspond à un primitif sémantique appelé « pensée » et désignant, généralement, les fonctions supérieures — cognitives dans une terminologie actuelle — de l'humain. Ce même radical se retrouve dans le terme latin *mens*, qui signifie « esprit » ou « mental ». Littéralement, le terme automate désigne donc un *dispositif capable de se contrôler par lui-même*.

Cependant, suivant l'interprétation la plus souvent rencontrée du radical, celui-ci se rattache à un autre primitif sémantique, appelé « mouvement ». Dans cette optique, l'automate devient *quelque chose qui se meut de lui-même* et, à ce titre, désigne



toutes sortes de machines. Cette interprétation, qui fait de l'automate un dispositif doué de mouvements, est à rejeter, parce que d'acception trop large. En effet, toute machine munie d'un moteur et d'un approvisionnement en énergie répond à cette définition; et pourtant, un tour ou une perceuse ne constituent pas, intuitivement, des automates. L'interprétation du terme automate adoptée ici sera plus restrictive. Il désigne une *classe particulière* de machines, à savoir des machines dotées d'une relative autonomie qui se gouvernent elles-mêmes. Leur autonomie implique à la fois une maîtrise du temps par la régulation et un contrôle du mouvement par la programmation. Si la régulation a été très tôt maîtrisée, il a fallu un peu moins de deux millénaires pour passer à la programmation.

2.1.3. Machine régulée

En Occident, la première machine dotée d'un dispositif de régulation recensé est celui des *clepsydras*, ou horloges à eau. Pour tenir le temps, la mécanique devait transformer une énergie de force irrégulière, ici le débit de l'eau, en un mouvement régulier.

Leur invention, au 3^e siècle avant notre ère, est attribuée au mécanicien Ctésibios (300-270). La description qui nous en est parvenue (près de deux siècles plus tard) est signée par Vitruve (1^{er} siècle avant notre ère), ingénieur et architecte romain. Ce dispositif est constitué d'un pointeau qui s'intercale dans une cavité intermédiaire entre la canalisation et le cylindre, qui se remplit d'eau de façon à pallier aux variations du débit d'eau. Un jour de l'an 246 avant J.-C., le même Ctésobios détourne un petit instrument à vent de son usage. Il s'agit de l'*aulos*, instrument de musique à anche double, qu'il fait fonctionner avec de l'eau. Il construit ainsi l'ancêtre de l'orgue, ingénieuse machine à faire de la musique, avec une pompe, des contrepoids, des pistons, des soupapes et une rangée de 8 à 10 aulos de différentes tailles, qu'il plante sur un sommier, c'est-à-dire une caisse rectangulaire en bois percée de trous sur le dessus pour y implanter les tuyaux, qui recevaient l'air par pression hydraulique.



Le *Théâtre d'automates* de Héron d'Alexandrie prouve une maîtrise remarquable dans l'art de réguler l'énergie motrice et de la transmettre entre divers composants. Ce dispositif à spectacle met en scène un drame mythologique complexe, la légende de Naupios et le retour des héros de Troie dans leur patrie. Les mouvements des personnages sont coordonnés aux changements de décor. Ce théâtre consiste en une machine actionnée par un mécanisme dont la force motrice repose sur le principe de poids et contrepoids, ceux-ci étant constitués avec du sable ou des masses de métal. Il utilise un arbre à came, c'est-à-dire un axe qui reçoit un mouvement de rotation et qui le transforme en mouvement de translation, ce qui permet de propager un même mouvement de diverses façons vers une vingtaine de statues, qui se transforment ainsi en personnages animés, représentant des hommes ou des dieux. Les organes moteurs des personnages sont constitués de leviers, de chevilles, de roues, de barrettes et d'un cylindre qui assure le contrôle de l'ensemble. Chaque crantage du cylindre engendre pour la composante associée un mouvement spécifique. Héron d'Alexandrie a mis au point la *rupture de séquence*, dispositif de régulation particulier qui permettait d'assigner des mouvements discrets (non-continus) aux personnages. Toutefois, cette machine spectaculaire ne répond qu'à moitié à la définition de l'automate. Certes, elle effectue des séquences d'opérations enchaînées sans aucune intervention humaine. Mais d'un autre côté, elle n'est pas dotée d'un organe de commande indépendant du mécanisme lui-même.

2.1.4. *Machine à programme*

Il a été brièvement question, dans le chapitre précédent, des machines à programme, au moment de traiter, plus généralement, (de l'imbrication) des diverses technologies qu'impliquent les médias interactifs (cf. section 1.2.8). Maintenant, on va revenir plus en détails sur la notion de programme dans le contexte particulier de l'automate.

Stricto sensu, un automate est une *machine endomécanique*, au sens où cette machine, pour constituer un automate, doit avoir en elle les lois de fonctionnement qui lui sont propres et doit les gérer. Ces lois, qui constituent un programme, sont



inscrites sur un support et mises en œuvre à partir d'un dispositif indépendant du mécanisme lui-même. En décodant le programme, ce mécanisme fournit à la machine la séquence des opérations enchaînées à effectuer. En distinguant, à l'intérieur du mécanisme, la description des différents mouvements d'une part et leur contrôle d'autre part, c'est-à-dire la chaîne causale qui les active ou les arrête, on opérerait un saut très significatif vers une plus grande abstraction de la machine et, du coup, vers l'élaboration d'une machine abstraite.

Les différents mouvements que l'on veut voir effectués par la machine doivent d'abord être répertoriés, puis standardisés. L'inventaire ainsi produit doit être décrit en terme d'entités dotées de paramètres dont la valeur varie sur des échelles données. Les opérations débouchant sur tel ou tel résultat consistent à transformer la valeur de l'un ou l'autre paramètre d'un objet.

Cette description est ensuite inscrite sur un support indépendant de la nature du dispositif qui l'applique, tel le carton perforé pour le métier à tisser, le tambour à picot pour les boîtes à musique, et, aujourd'hui les suites d'instructions pour les ordinateurs. Lors de son exécution, le programme est lu, décodé, et le mouvement ou l'action correspondant à chacune des instructions est effectué.

Les machines à programme se distinguent les unes des autres selon que la chaîne causale de leurs mouvements ou de leurs actions est *simple* ou *multiple*. Les métiers à tisser et les boîtes à musiques illustrent le premier type de machine, car la chaîne causale de leurs mouvements est dictée par la simple consécution des instructions. Bien sûr, on peut changer le programme, et le métier à tisser produira un motif différent, la boîte à musique interprètera une autre mélodie. Mais aussi bien le motif du tissu que la mélodie sont fixes et inamovibles. La chaîne causale peut cependant devenir multiple, et ce de deux façons. Soit le résultat d'une certaine opération peut donner des résultats différents, et l'action suivante varie avec le résultat sélectionné. Soit plusieurs chaînes causales différentes interviennent en parallèle ou de manière entremêlée.

Les automates du second type doivent être dotés de certaines capacités spécifiques. Premièrement, ils doivent être capables d'effectuer une sélection devant une



alternative. Deuxièmement, ils doivent pouvoir influencer sur l'ordre d'exécution des opérations, notamment en déclenchant certaines opérations sur des objets, en les répétant, ou, au contraire, en évitant la répétition dans des conditions données. Pour réaliser de tels automates, il a fallu réaliser un saut conceptuel dans la complexité structurelle du programme. Dans le cas du métier à tisser, les lignes du carton perforé guident les changements de position des fils de la trame un instant après l'autre, position qui varie en fonction de l'emplacement des trous sur la ligne courante. C'est la perforation ou l'inscription qui déclenche directement l'opération. L'enchaînement des lignes de perforation du carton qui avance décrit une seule chaîne causale. Le saut conceptuel dont il est question consiste à passer de l'enregistrement de l'opération par inscription sur un support à l'*information*, c'est-à-dire à la *description symbolique* des opérations et de leur enchaînement à l'aide d'un langage.

L'expression *description symbolique* renvoie à la notion de langage. Les instructions, ce sont des énoncés bien formés qui dictent l'accomplissement d'une opération. Parmi les instructions, il y a des instructions spéciales qui portent sur le contrôle des opérations. Plus complexes, ces instructions sur le contrôle des opérations sont de deux types : *instructions conditionnelles* ou *instructions itératives*. Les instructions conditionnelles ne seront exécutées *qu'à condition que* les conditions stipulées soient réunies. Les instructions itératives seront exécutées répétitivement jusqu'à ce que soient réunies les conditions d'arrêt, spécifiées par une condition ou par un échappement, c'est-à-dire une condition d'interruption du cycle courant ou une interruption complète.

Un assemblage d'instructions accomplissant une tâche donnée constitue un *algorithme*. Ce terme, emprunté aux mathématiques, désigne une succession *finie* d'opérations élémentaires à effectuer pas à pas, suivant un enchaînement strict, réalisée pour résoudre les problèmes d'une certaine classe. Pour décrire les procédés calculatoires des algorithmes — les *fonctions* — et les quantités objets du traitement qu'impliquent ces fonctions — les *variables* — il faut établir un registre



fini de symboles qui désignent les données, et une séquence d'instructions qui leur sont appliquées dans des conditions spécifiques. Le tout constitue le programme.

2.1.5. Temporalité

L'automate s'inscrit dans une temporalité qui distingue trois périodes : *stabilité*, *instabilité* et *métastabilité* (ou *retour à la stabilité*). L'environnement est stable pour l'automate lorsqu'aucun des paramètres auxquels l'automate est sensible n'est présent, ou lorsqu'aucun d'entre eux ne varie de manière perceptible. On entre dans une période d'instabilité lorsque l'automate réagit, que le programme est exécuté, ce qui peut modifier l'environnement. Lorsqu'il y a effectivement modification de l'environnement, on parle de métastabilité, ce qui est un peu l'équivalent, en quelque sorte, d'une stabilité " à valeur ajoutée ", puisqu'il y a eu modification. Si, au contraire, aucune modification n'est opérée sur l'environnement, il y a simplement retour à la stabilité.

2.1.6. Déterministes ou auto-adaptatifs

Suivant les relations qu'elles entretiennent avec leur environnement, les machines à programme que sont les automates relèvent d'une de ces deux classes *automates déterministes* ou *automates autoadaptatifs*.

Les premiers sont dits déterministes parce que leur programme prévoit l'ensemble des opérations qui seront effectuées *une fois pour toutes*. Alors, le programme est composé d'un ensemble de règles qui suffit à déterminer la conduite de l'automate pour effectuer une tâche donnée, en toute occasion et sans laisser aucune place à l'incertitude. Les automates déterministes sont par essence *discontinus*, parce que le cycle de leur fonctionnement a toujours un début (impliquant les matériaux ou les informations de base) et une fin (un produit ou une réponse). Une fois la tâche accomplie, une nouvelle tâche du même genre peut être lancée. Le programme décrit alors une succession régulière et harmonieuse des stades et des phases de fonctionnement pour former un cycle complet. Le *cycle* est le principe de régulation qui garantit la cohérence fonctionnelle des mouvements appliqués aux données de façon itérative. Un cycle complet peut être composé d'une succession ininterrompue



de cycles élémentaires et variés. Ainsi, par exemple, l'automate qui se cache derrière un guichet de banque automatisé est déterministe : le plus grand cycle commence avec l'insertion de la carte et se termine par le retrait ; ce grand cycle en contient d'autres, le premier correspondant au traitement du mot de passe qui, une fois réussi, autorise autant de transactions que le permet le solde du compte, constituant ainsi autant de cycles.

Le processus opératoire des automates adaptatifs (seconde classe d'automates) est *continu*, et leur programme vise à assurer un fonctionnement stabilisé produisant un résultat identique à deux instants successifs et à tout moment. Le cycle de ces automates repose sur la *rétroaction des actions*. Celle-ci confère à l'automate la capacité de réagir, sans intervention humaine, de manière adaptée aux variations de l'environnement dans lequel il se trouve inséré. Les règles du programme ont pour but de *maintenir*, par une action appropriée, une certaine grandeur à un degré ou une valeur donnée, en comparant sa valeur réelle avec une valeur qui est fixée d'avance et qui constitue, en quelque sorte, un point d'équilibre. Lorsque l'écart devient manifeste, un moteur secondaire est activé pour annuler cet écart. Ce type d'automate, dont la régulation est basé sur la rétroaction (c'est-à-dire le *feedback*), se retrouve, entre autres, dans le dispositif de pilotage automatique des avions.

2.1.7. Un système informatique abstrait

En informatique, le terme *automate* a été utilisé pour désigner une classe très précise de systèmes, capables de se placer par eux-mêmes dans un certain nombre d'états en fonction de règles. La structure formelle dynamique de ces systèmes a été initialement élaborée par A. Turing (cf. section 2.5.1). Ces systèmes répondent pleinement aux critères d'appartenance à la classe des automates définis plus haut. Pour réaliser un processus, un automate doit d'abord être *déclaré* au moyen d'une notation dans un certain langage de programmation et ensuite, le texte ainsi produit doit être exécuté sur un ordinateur dans l'environnement approprié. Une opération d'analyse est nécessaire pour découper le processus à informatiser en une séquence d'opérations ou en sous-processus, plus simples, qui constitueront les « états » de l'automate, respectivement $E_1, E_2, E_3, \dots, E_n$. Un *état* comporte 1) un ensemble de



valeurs admissibles, 2) une ou plusieurs actions caractérisant le sous-processus et, éventuellement, 3) une fonction de transition qui détermine un changement d'état. Un état représente un sous-processus dont le contrôle repose sur la sélection d'une action ou d'une opération, fondée sur la réalisation de l'une ou l'autre condition stipulée. La réception d'une donnée provenant de l'environnement provoque une réaction de l'automate: celui-ci change d'état, et, facultativement, produit une sortie. Voici un exemple d'automate composé de trois états et appliqué à l'analyse grammaticale.

S1	si {"le","la","les"}	-> CAT(article défini) et GO(S2)
	sinon si {"un","une","des","d"}	-> CAT(article indéfini) et GO(S2)
	sinon	-> GO(S1 D)
S2	tous	-> GROUPE(droite) et GO(S3 D)
S3	si {NOM}	-> GROUPE(gauche) et CAT(groupe nominal) et GO(S1 D)
	sinon	-> GO(S1 D)

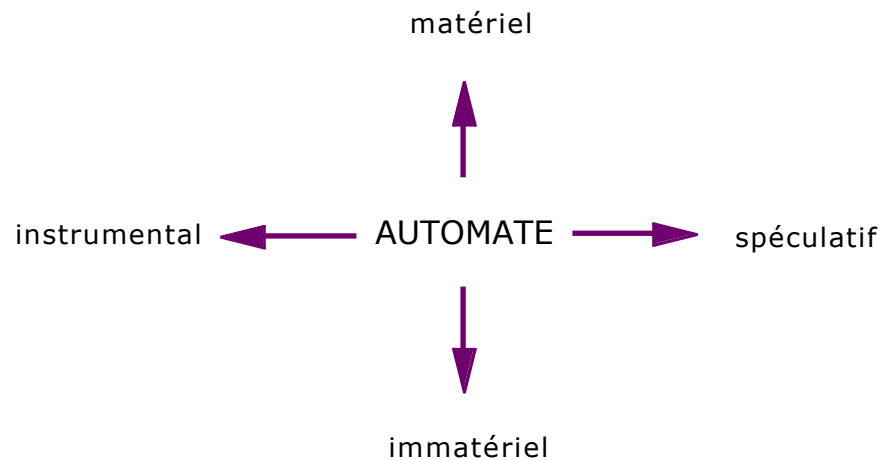
Un texte est découpé en mots, et l'automate les passe un à un. Lorsqu'il trouve un article, il lui accole une étiquette, ouvre un groupe et passe au mot suivant. S'il ne trouve pas d'article, il passe quand même au mot suivant. Si le mot suivant est un nom, alors il ferme le groupe précédemment ouvert et lui accole une étiquette. Sinon, il passe au mot suivant.

En tant que système abstrait, l'automate se retrouve aussi, comme on le verra plus bas, à la base même de l'ordinateur de Von Neumann (cf. section 2.5.1).

2.2. Approche instrumentale



Dans une approche instrumentale des automates, le terme *automate* désigne une



classe particulière de machines, à savoir celles qui sont gouvernées par elles-mêmes. De telles machines sont hors de la portée des techniciens de la Grèce antique. Il faudra attendre plus de deux millénaires avant de pouvoir attribuer pour la première fois le statut d'automate à des machines concrètes. Les *jaquemarts*, ces personnages animés couplés à des horloges, sont les premiers à qui l'on peut octroyer ce titre. Cela dit, les éléments et principes mécaniques primordiaux de l'automate ont déjà été systématisés par les maîtres de l'École d'Alexandrie, qui auraient réalisé des sculptures animées fort complexes. A côté des jaquemarts, on retrouve aussi, parmi les machines à automate, les boîtes à musique et les machines à calculer. Les boîtes à musique ont fourni les bases d'un théâtre de machines. Les machines à calculer, dont on dérivera des machines à penser, ont jeté les bases de l'ordinateur, le support par excellence des médias interactifs. Cette idée de machines à penser sera au cœur du projet de l'intelligence artificielle, qui a connu son apogée dans les années 1980. L'idée de reproduire le raisonnement humain et même la vie avec un ordinateur transformera l'automate instrumental en un automate spéculatif.

2.2.1. L'horloge



L'horloge est une machine auto-régulée qui, en vertu de son principe interne, transforme l'énergie en information. En tant que telle, elle peut prétendre davantage au titre d'ancêtre des automates instrumentaux que les théâtres de machines.

Le cadran solaire, connu depuis la préhistoire, permettait de connaître l'heure directement en observant soit la longueur de l'ombre du soleil produite par un stylet vertical, le *gnomon*, soit la direction de cette ombre sur le cadran. Les clepsydres, apparues en Égypte environ 3 000 ans avant notre ère, sont des horloges qui fonctionnent à l'eau. Les sabliers, couramment utilisés au 14^e siècle, mesurent le temps par l'écoulement d'un fluide, tout comme les horloges à eau. Ces appareils ne permettent pas de déterminer l'heure, mais servent à mesurer une certaine durée.

L'exactitude constante de la mesure du temps est assurée par la maîtrise du *principe de régulation*, à l'aide du pointeau des clepsydres, ces horloges à eau antiques, ou du pendule qui oscille dans les premières horloges mécaniques (nées au 6^e ou au 10^e siècle selon les sources). Le dispositif de régulation de l'horloge a été généralisé dans le *principe de l'échappement*, introduit au 14^e siècle. Tout mouvement connaît sans cesse des inégalités dues aux conditions matérielles de sa production, qu'il soit initié par l'eau qui s'écoule, par un poids qui descend ou par un ressort en spirale qui se déroule (force motrice introduite au 17^e siècle). Le *régulateur* transforme le mouvement rotatif *continu* mais *irrégulier* en un mouvement *alternatif* et *régulier*.

L'horloge moderne constitue un dispositif entièrement mécanique où l'énergie, au lieu de provenir de l'écoulement continu d'un fluide, est produite par le mouvement discontinu d'un rouage à roues et pignons dentés. Ce mouvement est produit par un poids soumis à la gravité, ou par un ressort enroulé en spirale. Ce poids ou ce ressort fournit de petites impulsions discrètes à un organe, *l'échappement*, qui agit lui-même sur un *régulateur*, le pendule.

L'impact des horloges sur la vie des gens fut très grand, aussi bien sur le plan de l'organisation des activités (comme les activités de production) que sur le plan de la philosophie. Tout comme l'ordinateur à notre époque, l'horloge influencera de



manière capitale, en son temps, la représentation que l'homme a de lui-même et de son rapport au monde environnant. La pensée de René Descartes témoigne particulièrement bien de cette influence.

2.2.2. *Le calculateur arithmétique*

Les premiers calculateurs s'appliquent à l'arithmétique. Ce sont des machines qui effectuent pour l'humain de longs et fastidieux calculs, similaires et répétitifs, qu'il était nécessaire de faire pour déterminer la teneur des impôts à percevoir par l'état, ou encore pour déterminer des tables de navigation, etc. La mécanique de ces machines reproduit le principe technique du boulier. Pour concevoir ces machines, il fallait, au préalable, synthétiser l'ordre temporel des opérations de base du calcul.

En 1624, l'allemand Wilhem Schickardt (1592-1635) invente une « horloge à calculer » pour offrir une aide à son maître, l'astronome Johannes Kepler (1571-1639). Cette horloge à calculer comporte six tables de multiplication complètes et indépendantes et, surtout, une machine à effectuer des additions, constituée de six disques verticaux placés face à l'opérateur. Ces six disques qui représentent respectivement de droite à gauche les unités, les dizaines, les centaines et ainsi de suite, sont animés par cinq roues intermédiaires à dix dents. Chacune de ces roues est mise en mouvement par une dent de la roue précédente, située entre le 9 et le 0, ce qui provoque une rotation d'un dixième de tour de la roue suivante vers la gauche. Pour effectuer une soustraction, il fallait inverser la procédure. Un seul exemplaire de cette machine est supposé avoir été construit. Il a été détruit six mois après.

Le « totalisateur » (1642), appelée aussi pascaline de Blaise Pascal (1623-1662), est une machine capable d'effectuer des additions avec retenue automatique. Au moment où une roue dépasse le 9, un levier fait bouger la roue suivante. Cette machine, plus compliquée que la précédente, a été plus largement diffusée: plus d'une cinquantaine d'exemplaires ont été fabriqués.

Gottfried Wilhem baron de Leibniz (1646-1716) a travaillé à la conception d'une machine qui pourrait calculer automatiquement les tables trigonométriques. Il



invente le tambour à dents inégales, qui remplaçait à lui seul dix rouages élémentaires. Un exemplaire de cette machine construit en 1694 a été conservé, et un autre, construit en 1704, a été perdu.

Charles Stanhope (1753-1816) contribue de deux façons aux machines à calculer. D'une part, il propose en 1775 une machine à calculer dont un exemplaire aurait été possédé par Charles Babbage. Cette machine effectue les quatre opérations avec simplicité, fiabilité et facilité d'emploi. D'autre part, il a travaillé pendant près de trente ans sur la première machine logique connue au monde, le « demonstrator ». Cette machine était destinée à résoudre certains syllogismes classiques et des problèmes élémentaires de probabilité. Il n'y a que des témoignages indirects de cette machine, qui est un jalon très important dans la mécanisation de la pensée. En effet, c'est la première trace d'une machine dont le calcul serait non pas arithmétique, mais logique ou symbolique. Cette machine n'opère pas sur des valeurs numériques, mais sur des *valeurs symboliques*, qui sont cumulées et déduites, mais aussi comparées et inscrites dans une chaîne d'implication de propositions. Chacun des termes de la proposition est introduit par une particule. Les prémisses dont la valeur doit être fixée sont introduites par SI, SINON, ou SINON SI. La conclusion est introduite par ALORS. L'opération logique de l'implication consiste, une fois que la valeur de vérité des valeurs ou des opérations de la prémisse est vérifiée et avérée, à transférer cette valeur de vérité sur la conclusion de l'énoncé. L'énoncé dont on a démontré la vérité est ensuite utilisé pour analyser d'autres énoncés, et ainsi de suite, jusqu'à ce l'on arrive, par décompositions successives, à l'énoncé final qu'il fallait démontrer.

La *Machine à calculer à différence*, proposée en 1822 par Charles Babbage (1791-1871), appartient à la première génération de calculatrices capables d'effectuer automatiquement des séquences d'opérations enchaînées, suivant un processus fixé d'avance, et exécuté par un mécanisme de commande. Appelée « automate à programmation unique », cette machine qui calcule toutes les opérations rapidement utilise la méthode mathématique des *différences finies*. Cette méthode consiste, à partir d'une valeur initiale donnée, d'effectuer le remplacement du calcul



des valeurs d'un polynôme de degré n par une série de n additions successives. Cette machine à calculer constitue le premier automate proprement dit, puisqu'il satisfait les deux conditions de complexité posées : elle est dotée à la fois d'un dispositif de régulation, et d'un dispositif de programmation.

Toutes les calculatrices passées en revue jusqu'à présent sont *analogiques*, adjectif qui signifie *proportionnel* en grec. En effet, ces machines ne travaillent pas sur des nombres à proprement parler, mais sur des grandeurs physiques : le déplacement d'une réglette, la rotation d'un ou plusieurs axes et, plus tard, la variation d'un potentiel électrique. Dans tous les cas, la machine matérialise un phénomène physique dont les variations, déplacements ou transformations mesurables évoluent selon une loi analogue à celle qui régit l'expression mathématique ou physique à traiter. La limite de ce type de stratégie, c'est qu'elle ne permet que de résoudre le type de problème pour lequel l'analogie fonctionne.

2.2.3. Le calculateur analytique

En 1834, Charles Babbage élabore l'idée d'un *calculateur analytique*, idée qu'il révisera sans cesse. Les calculateurs analytiques sont parfois appelés *automates intellectuels*. L'*analyse*, c'est l'action de délier, de décomposer un tout en ses parties, et de décomposer encore les parties en parties élémentaires. Le calcul analytique exige, tout d'abord, de déterminer les règles et les procédures qui permettent de résoudre un problème. L'analyse consiste à reformuler ces règles plusieurs fois, pour obtenir une formulation exécutable sous la forme d'un programme qui décrit les séquences d'instructions correspondant à chacune des étapes élémentaires de la procédure. Celle-ci doit être exécutée automatiquement.

La machine analytique de Babbage peut non seulement effectuer n'importe quel calcul, mais peut en outre organiser son travail en fonction des instructions données. L'organisation des registres mécaniques n'est plus considérée comme linéaire et les valeurs à calculer ne découlent donc plus de leur position.

L'architecture de la machine analytique, dès 1840, a déjà la structure de base de l'ordinateur contemporain. Elle distingue déjà la fonction « calcul », la fonction



« mémoire » et la fonction « contrôle des opérations ». Cette dernière fonction, qui constitue, en quelque sorte, une faculté de décision, effectue du « branchement conditionnel ». Autrement dit, la machine est capable de sélectionner seule le terme d'une alternative qui doit être retenu, soit à partir d'une règle, soit en fonction du résultat des calculs précédents. Cette fonction permet aussi de répéter n fois une instruction ou une série d'instructions. En outre, il était prévu qu'un programme annexe soit appelé et exécuté avant le retour au programme principal.

La machine analytique de Babbage maîtrise pleinement la régulation. La notion clé de cette machine, c'est le cycle élémentaire, défini comme la plus petite unité de temps nécessaire pour passer d'un chiffre décimal au chiffre qui le suit immédiatement sur un même rouage. À partir de cette unité de temps arbitraire, il était possible d'estimer le nombre de cycles nécessaires pour l'exécution des opérations simples puis, par cumul, le nombre de cycles nécessaires pour exécuter un algorithme, programme composé d'un assemblage de séquences d'opérations simples. Ces opérations utilisent la mémoire pour stocker les résultats qui seront ensuite rappelés lors d'opérations ultérieures.

Cette machine regroupe plusieurs unités périphériques autour de l'unité qui effectue les calculs:

- 1) l'unité d'alimentation en nombres et en instructions,
- 2) l'unité de stockage des résultats partiels, fonctionnant comme une mémoire à court terme
- 3) un dispositif de contrôle qui compare les résultats partiels pour prendre des décisions sur la suite des opérations à mener
- 4) un magasin, mémoire à plus long terme.

Cette machine ne sera jamais construite entièrement, car la mécanique ne permettait pas d'atteindre la précision exigée, et entraînait un frottement qui l'aurait empêché d'atteindre une vitesse de traitement acceptable; Babbage nous a néanmoins légué de six à sept milles pages de notes sur cette machine.



Par ailleurs, en 1869, William Stanley Jevons (1835-1882), économiste britannique, propose un « piano logique », première machine capable de battre l'humain en termes de vitesse sur des problèmes logiques précis. Cette machine incorpore les principes logiques développés par George Boole. Plus précisément, elle dérive de l'*abaque*, planchette à calculer utilisée depuis l'Antiquité aux côtés du boulier. Au lieu de comporter des nombres, les planchettes rectangulaires (de taille identique) portaient une des deux valeurs « 1 » et « 0 », pour exprimer respectivement « vrai » ou « faux ». Le piano permettait d'effectuer des opérations logiques sur des combinaisons différentes de termes vrais ou faux, et d'afficher le résultat. Ainsi, on frappait les touches dans l'ordre des termes de l'équation logique, et, en fin de travail, la machine affichait la valeur de vérité finale, résultant de l'opération complète.

2.2.4. *Calculateurs électromagnétiques ou électroniques ?*

On l'a vu, la machine analytique de Babbage était impossible à réaliser à cause de la friction dans les rouages de la mécanique. Pour résoudre ce problème, deux stratégies différentes ont été poursuivies : les *relais téléphoniques* basés sur l'électromagnétique, et les *tubes électroniques* qui, avec un temps de réaction de 200 000 pulsations à la seconde, sont mille fois plus rapides que les relais. Le premier *totalisateur électrique* est dû à George Stibitz (1904 - 1995) des laboratoires Bell aux Etats-Unis en 1937. Il s'inspire de certaines techniques téléphoniques où des relais sont utilisés pour constituer des circuits binaires. Plusieurs générations de cette machine robuste ont été réalisées et utilisées massivement durant la Seconde Guerre mondiale, notamment pour la conduite de tirs antiaériens.

Achévé en 1944 à l'Université de Harvard aux États-Unis, l'*Automatic Sequence Controlled Calculator Mark 1* est une machine dotée d'un organe de calcul fondé sur les relais téléphoniques. Mais sa mémoire, composée de roues, demeure mécanique. Cette machine aux proportions gigantesques pesait 5 tonnes et comprenait 765 299 pièces différentes : 863 kilomètres de fils électriques, un millier de commutateurs et 2 200 relais à 10 positions ; 1 210 roulements à billes, 175 000 connexions



électriques et 3 millions de points de soudures. Animé par un moteur de 5 chevaux, un arbre primaire horizontal courait à la base de la machine sur toute sa longueur et en assurait la régulation. L'entrée des informations se faisait par cartes ou bandes perforées. Cette machine, qui a vécu quinze ans, était branchée sur le réseau téléphonique et pouvait ainsi être activée à distance.

Achévé en 1945 à l'Université de Pensylvanie, l'*ENIAC — Electronical Numeral Integrator and Calculator* — sera le premier calculateur analytique multifonction électronique. L'ENIAC, aussi gigantesque que l'ASCC, occupait une surface de 200 m² avec ses 42 armoires de tôle d'acier peinte en noir ; chacune d'elles mesurait 3 mètres de hauteur, 60 cm de largeur et 30 cm de profondeur. La chaleur dégagée par les 18 000 tubes était telle que deux gros moteurs Chrysler de 12 CV tiraient l'air chaud au moyen de conduites d'aération placées sur le haut des armoires. La machine consommait près de 200 kilowatts de courant électrique. Trois armoires, plus larges, montées sur des roulettes, pouvaient être connectées à divers endroits de la machine selon la nature du travail requis. Un lecteur et un perforateur complétaient le calculateur. Pour changer le programme de calcul, il fallait modifier la position des commutateurs à même les armoires et modifier les lignes de liaison entre armoires au moyen de fiches dans un tableau de connexion. L'ENIAC a frappé l'imagination de ses contemporains pour deux raisons. Tout d'abord, il formait le plus vaste assemblage électronique réuni en une même machine, au moins cent fois supérieur en volume aux montages antérieurs. Ensuite, sa vitesse de calcul était prodigieuse, et ce en dépit de la fragilité des lampes qui grillaient fréquemment, et qui provoquaient donc des pannes. On rapporte qu'entre le premier et le cinquième mois d'exploitation, le nombre d'ampoules qu'il fallait changer est passé de 50 à 15 par mois et que, par la suite, une lampe brûlait tous les deux jours en moyenne.

Il a fallu attendre l'ordinateur pour obtenir une machine digitale logique qui intègre les deux composantes essentielles de l'automate, à savoir la régulation des opérations, et leur programmation.

2.2.5. *L'automate industriel*



L'automatisation ne s'est pas bornée au domaine des calculateurs, dont l'importance est cependant prépondérante, vu qu'ils mènent tout droit à l'ordinateur. Rappelons les quatre phases successives de mécanisation qui mène à l'automate industriel que nous connaissons aujourd'hui.

Lors d'une *première phase de mécanisation* qui prend place vers la fin de l'Antiquité, les machines ont remplacé les outils. Les machines sont constituées d'ensembles de composants, animés à partir d'un même mouvement, fourni par une force motrice et formant des chaînes cinétiques intégrées. Les mécanismes des machines accroissent à la fois la puissance de l'outil, le rendement du travailleur humain et la standardisation du produit. Parmi les premières machines d'usage industriel, on compte le *tour métallique à chariot* introduit en 1751, le *tour à aléser*, qui permet de construire des cylindres pour les machines à vapeur dès 1775, et le *tour à fileter* qui permet, à partir de 1795, de fabriquer les écrous nécessaires à l'érection de structures métalliques.

L'*industrialisation* correspond à la deuxième phase de mécanisation intensive et globale. Avec l'industrialisation apparaissent les usines, lieux physiques de plus en plus imposants. Ces usines contiennent l'ensemble des machines nécessaires à la fabrication d'un produit, à la transformation de la matière première et des produits semi-finis en produits finis. Elles produisent ou transforment la force motrice nécessaire au fonctionnement des machines. L'industrialisation désigne la *mécanisation généralisée*, c'est-à-dire la prise en charge par les machines de l'ensemble des procédés de fabrication d'un produit donné.

L'étude de ces machines a mené à la *cinématique*, science qui permet de réduire les machines à leurs mouvements principaux, indépendamment de la force qui les anime : eau, vapeur, électricité, essence. Une fois ce savoir acquis, le mouvement inverse peut être effectué pour concevoir une machine en distinguant de manière linéaire et arborescente les tâches à accomplir, et en identifiant, pour chacune de ces tâches, les constituants et les mouvements qui forment une chaîne cinétique pouvant être coordonnée à d'autres chaînes, de manière à accomplir des tâches plus complexes. C'est ainsi que petit à petit, les techniques ne sont plus classifiées selon



les produits qu'elles permettent de réaliser, mais selon les opérations qu'elles permettent d'accomplir. La *réduction théorique des figures mécaniques* constitue l'un des principes majeurs de la technologie moderne.

La troisième phase de la mécanisation, c'est la phase d'*automatisation*, au cœur de la révolution industrielle. Les machines sont dotées de la capacité de réaliser certains automatismes. Le travailleur n'a plus à réaliser la tâche, il n'a plus qu'à s'occuper du flot : il se borne à alimenter les machines en matière première, et à en recueillir le produit pour alimenter une autre machine, et ainsi de suite. La production de biens en série par des machines automatiques permet de produire une quantité de plus en plus considérable de produits, dans un laps de temps de plus en plus court, et au moindre coût. L'automatisation implique une *normalisation* et une *standardisation* des produits. Par exemple, vers 1835, apparaît l'idée de pointures pour les chaussures. Le passage de machines semi-automatiques aux machines automatiques s'est opéré vers 1880, avec l'emploi généralisé d'automates adaptatifs. Ceux-ci constituaient des boucles de commandes automatiques, opérant à partir de niveaux de température, de pression, de concentration ou encore d'autres variables physiques ou chimiques. Cette phase d'automatisation assure le passage de la production individuelle à la production de masse. Toutefois, les différentes boucles de commandes automatiques fonctionnent indépendamment les unes des autres, selon un programme pré-établi. Leur coordination exige de nombreuses opérations de contrôle et d'autres interventions humaines.

La quatrième phase de mécanisation, beaucoup plus récente, est celle de *l'automation de la production* par l'intégration d'un certain nombre d'automatismes. Il y a automation lorsque les opérations d'analyse, d'organisation et de commande des machines sont prises en charge par des machines. Jusqu'ici, ces opérations étaient effectuées par des travailleurs. Le summum de l'automation, c'est *l'usine sans travailleurs*. Dans les années 1955-1960, le contrôle automatique, jusque là intégré à chacune des machines, est étendu aux opérations de commande et de coordination, à partir de schémas imposés. L'automate assure la coordination d'automatismes préalablement définis, supplée aux ruptures d'automatisme, ou, du



moins, produit un diagnostic de la panne. Enfin, il gère les échappements, c'est-à-dire décide du moment où le processus est terminé. L'automatisation implique l'alimentation automatique des machines. La liaison entre les machines automatiques est assurée par des palpeurs de circulation ou d'écoulement des matières. L'automatisation a permis d'automatiser les cycles successifs de contrôle de la qualité : diagnostic des anomalies et application de corrections adaptées.

Cette révolution fait voir la production sous un tout nouveau jour: l'homme et la machine peuvent échanger leur fonctions. En même temps, elle asservit l'homme qui se trouve désormais intégré à des circuits automatiques sur lesquels il n'exerce pas de contrôle, son rôle se limitant à répéter la même opération. Il ne constitue donc qu'un automatisme parmi d'autres, que ceux-ci soient assurés par des machines ou d'autres travailleurs. Les travaux de Frederick Winslow Taylor (1856-1915) vont un peu dans ce sens. Ces travaux promeuvent l'organisation scientifique du travail, en proposant, en quelque sorte, de mécaniser les travailleurs pour les rendre adéquats aux machines.

2.3. Approche spéculative

On l'a dit plus haut, l'automate n'a pas toujours une finalité instrumentale. Il peut aussi avoir une fonction *spéculative*. Le terme " automate " désigne alors les artefacts qui font l'objet de mythes ou de spéculations dans l'imagination populaire. Parmi ces mythes figure au premier plan le mythe originel de la fabrication d'un androïde par l'homme, qui lui conférerait la vie. Des automates spéculatifs apparaissent déjà dans les récits mythiques, entre autres sous la forme de statues automatiques parlantes. Il s'agit, en fait, des marionnettes qui tiennent un rôle d'oracle, d'intermédiaire entre les dimensions mythique et physique de la vie. La tête d'Orphée à Lesbos qui aurait prédit la mort de Cyrus en est un bel exemple. Presque sans discontinuer, cette quête du double se prolonge jusqu'à l'époque contemporaine, pour déboucher, finalement, sur des automates digitaux. Ces automates sont dits « intelligents », parce qu'il peuvent raisonner à partir de faits, ou encore parce qu'ils peuvent comprendre et traduire les langues naturelles.



2.3.1. Le miroir

Le thème de l'automate a exercé une fascination transhistorique sur l'imaginaire l'humain. À côté des automates instrumentaux, dont l'utilité pratique est claire, il y a toujours eu des automates qui relèvent plutôt de la spéculation, réalisés ou seulement imaginés. Spéculation: miroir ou fantasme. Selon les époques et les publics, c'est l'un ou l'autre sens du terme qui prend le dessus. Dans les deux cas, l'automate pose la question de la nature de l'homme. À travers les automates, l'homme examine les contours de son humanité. Les automates spéculatifs ont aussi joué un rôle déterminant dans l'équilibre social et le progrès de la civilisation.

Les automates provoquent un effet miroir lorsque l'humain les utilise pour expliquer son propre fonctionnement. Cet effet miroir fut particulièrement fort à trois époques: l'Antiquité grecque, les 17^e et 18^e siècles, et ces dernières années. À ces trois époques, philosophes et hommes de science ont, de manière réductionniste, considéré l'humain et les êtres vivants un peu comme des automates. Les automates spéculatifs jouent alors une fonction heuristique essentielle : ils permettent de se reconnaître soi-même dans un autre, en ménageant une tension entre le rapprochement et la mise à distance.

Durant la première de ces époques (l'Antiquité), c'est principalement à travers Aristote que s'exprime cet effet miroir de l'automate. À partir des principes de base de la mécanique, Aristote conçoit le corps humain comme une convergence d'organes-outils différenciés et spécialisés, les cieux comme un agencement de sphères, et le principe vital comme un moteur. La deuxième époque (17^e et 18^e siècles), c'est l'époque de l'horloge et, donc, de la maîtrise du temps et du mouvement. Le mécanisme de l'horloge qui fascinait les penseurs est généralisé à la nature et à la vie. Le corps humain devient automate. René Descartes (1596-1650), dans le *Discours de la méthode* (1637), met en relation les sciences de la vie (médecine et biologie) avec la technologie. Descartes fait aussi jouer à l'automate le rôle de machine philosophique. L'automate devient le modèle explicatif des mouvements vitaux du corps humain auxquels la volonté ne participe pas. La démarche de Descartes consiste à traduire la vie en machines, en pièces, en



arrangements mécaniques, pour alors décrire et analyser la mécanique des fonctions vitales des organes.

L'automate est aussi une constante du système philosophique de Leibniz. Dans les *Monadologies* (1714), au numéro 64, Leibniz définit le monde comme une machine composée d'agrégats, et le corps vivant comme une machine divine, ou comme un automate naturel. Dans *L'homme-machine* (1747), Julien Offray de La Mettrie (1709-1751) donne au mécanisme cartésien une dimension psycho-physiologique, en proclamant la primauté de l'expérience médicale. Il fournit un modèle para-scientifique d'explication du monde, en faisant abstraction complète de la transcendance, c'est-à-dire sans recourir à aucune médiation théologique et cosmologique au-dessus ou en-dehors de la nature. C'est pour cela que ce modèle est dit « matérialiste ».

La troisième époque marquée par l'effet miroir de l'automate spéculatif, c'est l'époque des ordinateurs. Les scientifiques comme John von Neumann (1903-1957) s'inspirent de l'architecture neuronale pour développer l'architecture de l'ordinateur. Plus tard, d'autres scientifiques s'inspireront du raisonnement logique ou des réseaux neuronaux pour développer une intelligence artificielle.

2.3.2. *Le fantasme*

L'autre aspect de la spéculation, c'est le fantasme, construction imaginaire qui échappe aux contraintes de la réalité, voire de l'ordre établi. Le fantasme de l'automate, c'est la reproduction artificielle. Produire un double humain à l'aide des technologies, lui donner vie en contournant les modalités biologiques, sans intervention divine. Mimer l'homme par un artéfact ne suffit plus; le fantasme, c'est la création d'un androïde, d'un homme artificiel.

On retrouve ce fantasme à l'œuvre dans la mythologie antique: les hommes, les femmes et les automates sont l'œuvre du même démiurge. Héphaïstos, dieu du feu et des forges, donne la vie aux objets qu'il fabrique. Il crée des automates doués de vie, telles les servantes d'or qui aident le dieu estropié à se mouvoir, ou Pandora, figurine de terre cuite et Talos, homme de bronze affecté par Minos à la garde de la



Crète, dont il faisait le tour d'un pas régulier. Avec Athéna, les Dieux créèrent la femme; avec Prométhée, ils créèrent l'homme. Dédale fait aussi partie de l'histoire des automates en Grèce antique. Architecte et sculpteur, il construisit le labyrinthe de Crète, dans lequel fut enfermé le Minotaure et, finalement, Dédale lui-même. Celui-ci s'en échappa avec l'aide de son fils Icare, au moyen d'ailes composées de plumes et de cire. Dédale et Icare forment ainsi les principales figures de l'artisanat automatique hellénique, ce qui indique un glissement de la démiurgie magique à la création de dispositifs concrets. Pygmalion, jeune roi de Chypre, fabrique de ses mains une statue d'ivoire, Galatée, dont il tombe amoureux. Aphrodite, déesse de l'amour, sensible au désespoir du jeune homme, donne vie à la statue. La conséquence de cet acte est plutôt grave, puisqu'à cause de ce geste, la descendance du roi n'est plus assurée.

Parallèlement, on retrouve ce fantasme à l'œuvre dans l'alchimie et dans la Kabbale. L'alchimie désigne un corpus de textes et de pratiques qui prirent cours en Chine, dans les Indes, dans l'empire arabe et, enfin, dans l'Occident latin du Moyen-Âge et de la Renaissance, où l'alchimie a connu son apogée. La *philosophie hermétique* (attribuée à Hermès Trismégiste, littéralement " trois fois grand ") s'attache aux correspondances entre le monde extérieur à l'humain — le macrocosme — et les différentes parties du corps humain — le microcosme. Le *Grand Œuvre* est l'objet de la quête initiatique des alchimistes, quête aussi bien matérielle que spirituelle. Le Grand Œuvre consiste en l'inversion des rapports démiurgiques de l'homme au cosmos. La réussite du Grand Œuvre génère une force telle que le plomb, métal le plus impur, est changé en or, métal le plus pur. Cette force est réputée si grande que certains traités affirment que la vie de l'alchimiste devient éternelle ; d'autres traités affirment qu'elle peut donner vie à de petits êtres à forme humaine appelés *homoncules*. Les homoncules sont des nains monstrueux et forts d'une connaissance des choses secrètes. Ce terme, apparu au début du 17^e siècle dans un traité alchimique ésotérique attribué au médecin et alchimiste suisse Paracelse (1493-1541), servira à désigner, au 18^e, le petit être que les biologistes voyaient dans l'ovule ou dans le spermatozoïde (Buffon 1749).



2.3.3. *Ésotérisme*

Les automates spéculatifs ont fait l'objet de plusieurs traités qui nous sont parvenus en partie. Ils peuvent se définir comme des interprétations divinatoires ou magiques de traités techniques d'alchimie, de mécanique, de médecine et de numérologie. Ils se veulent ésotériques: leur sens caché est réservé aux initiés. Chez le lecteur, la quête de sens se double d'une quête de l'initiation, préalable nécessaire à la compréhension du sens caché. La quête du lecteur se transforme, du coup, en un cercle vicieux, puisqu'il ne peut être initié qu'une fois qu'il a compris le sens caché. C'est ainsi que sont apparus des traités périphériques aux disciplines mentionnées, souvent faussement attribués à des autorités comme Aristote, Albert le Grand, Thomas d'Aquin, Bacon, Lulle, etc. Fait d'initiés, ces traités fournissent les clés de l'initiation, mais en les cachant soigneusement à travers des métaphores, voire des allégories, ou encore à travers des transpositions de registre, et d'autres procédés similaires.

2.3.4. *Quelques jalons*

Dans le domaine de la création littéraire, théâtrale et cinématographique, le thème de la création d'un double humain fut extrêmement fécond. Voici quelques jalons particulièrement connus et marquants de cette histoire, qui n'épuisent pas, loin s'en faut, l'ensemble des textes ésotériques, des romans et des films consacrés à cette thématique.

La Kabbale — de l'hébreu *qabbala*, " tradition " — constitue un dispositif de réponses aux grandes questions communes à la religion et la philosophie. Elle prend la forme de figures conceptuelles ou imaginaires remplies de symboles.

L'interprétation de la Kabbale a exercé une grande fascination sur les philosophes, dont, entre autres, Spinoza ou Leibniz, ainsi que sur les écrivains comme Goëthe ou Schelling.

Le *Golem*, roman de Gustav Meyrink, reprend une légende suivant laquelle le rabbin Elijat de Chelm ou Yehudah ben Loew aurait fabriqué un homme artificiel en 1580, à



Prague. Par des incantations, il aurait réussi à faire entrer le Nom Sacré dans une statue d'argile, dont les yeux se seraient ouverts.

Le *Docteur Frankenstein* de Mary Shelley (1797-1851) raconte l'histoire d'un jeune étudiant en médecine obsédé par le désir de trouver le secret de la vie, et qui réussit à créer un être vivant artificiel.

RUR Rosum's Universal Robot est un drame d'anticipation du tchèque Karl Capek, représenté pour la première fois en 1921. L'action prend place dans le futur, sur une île du dénommé Rossum — du tchèque *rozum*, « raison » — où se trouve une fabrique d'ouvriers artificiels, de robots — du tchèque *robota*, « travail » — créés par le neveu de Rossum.

Le film *Metropolis* de Fritz Lang (1927) est un drame de science-fiction qui met en scène une femme artificielle. Toujours au cinéma, notons que le personnage le plus achevé du film *2001 l'odyssée de l'espace* de Stanley Kubrick (1968) est un ordinateur intelligent, HAL (ou IBM décalé d'une lettre), faisant preuve de plus d'émotions que les humains qui ont l'air de robots à côté de lui. Le film *Blade Runner* de Ridley Scott (1982) met en scène des *répliques*, êtres humains artificiels au service des humains, déployés dans les environnements les plus dangereux pour faire la guerre et ainsi coloniser d'autres planètes. Dans leur dernière incarnation Nexus 6, les répliques surpassent leurs créateurs, et sont alors mis hors la loi puis exécutés.

Toutes ces œuvres synthétisent les fantasmes oniriques et les réalités pratiques d'une époque. Contrairement aux artefacts décrits dans les sections suivantes, les créatures artificielles ont un caractère strictement imaginaire, même si elles empruntent aux technologies les plus avancées de leur temps. Par leur richesse symbolique, les créatures artificielles ont néanmoins, au même titre que les récits bibliques ou mythiques, une importance philosophique capitale, particulièrement sur *l'éthique*, en phase directe avec les développements scientifiques touchant à l'homme. Enfin, l'attraction exercée par les créatures artificielles, substitués d'hommes et de femmes, n'a pas uniquement un caractère rationnel; elle est surtout émotionnelle, voire carrément érotique.



2.3.5. Structure du récit mythique

Dans un tel corpus de récits de création d'êtres artificiels, constitué de mythes, de légendes, de romans, de films mais aussi d'articles scientifiques, l'anthropologue structuraliste recherche avant tout le *récit de base*, fondateur, *invariant* sous la diversité des contextes historiques et des modes d'expression. Créer un double humain, animer une matière inerte, lui conférer la vie, le mouvement et la pensée humaine, voilà bien un mythe transhistorique.

L'analyse proposée ici est celle que développe Philippe Breton dans un livre intitulé *À l'image de l'homme : du Golem aux créatures virtuelles* (1985). Breton scinde le récit fondateur de la création d'un être artificiel en trois parties distinctes.

Premièrement, la créature émerge de la combinaison d'éléments ou de matériaux, clairement identifiables, et, le plus souvent, *primitifs*, ce qui permet d'enraciner la créature dans la matière. Deuxièmement, la fabrication ou le modelage de la créature par l'homme, artisan ou savant, renforce la matérialité de la créature et transforme l'acte technique en un processus de création. Troisièmement, la créature doit prendre vie. L'entreprise de mise en forme par l'homme se révèle parfois insuffisante; une intervention extérieure à l'univers strictement humain est alors nécessaire, soit par l'intermédiaire d'un dieu, d'un savoir initiatique ou scientifique.

Ce mythe connaîtra plusieurs dénouements. Suivant le plus fréquent d'entre eux, la créature se retourne contre son créateur. Faut-il y voir une récupération judéo-chrétienne, une version du châtement infligé à celui qui outrepassé ses attributs d'humain ?

L'automate brouille la frontière entre réalité et imaginaire technique; il dramatise, condense et déplace les connaissances pratiques d'une époque afin de faire entrer la machine dans la sphère existentielle par artifice. Les sections suivantes relatent trois moments clés de l'histoire des automates : les masques et les statues antiques, les jaquemarts de la Renaissance, et les androïdes de l'époque classique.

2.3.6. Masques et statues antiques



Comprendre les médias interactifs de Louis-Claude Paquin est mis à disposition selon les termes de la licence Creative Commons 4.0 : Attribution - Pas d'utilisation commerciale - Pas de modification.

Le *masque articulé* et la *statue animée* constituent les plus anciennes manifestations de l'automate spéculatif. Rendant les mouvements limités et peu autonomes, le masque est surtout utilisé lors de rituels. Les rituels constituent des séquences de gestes déterminées, réalisées pour provoquer un certain résultat: favoriser la fécondité, conjurer un malheur, purifier, célébrer un événement. À ce titre, les rituels peuvent se concevoir comme des *automates symboliques*, dont le mécanisme est composé par les masques des participants. Tout comme les statuaires, les icônes ou les totems, les masques renforcent la cohérence du groupe, le sentiment d'identité collective propre à un groupe d'humains formant une communauté. Sont parfois utilisées lors de rituels des effigies de dieux sous forme humaine, faites d'argile et aux articulations commandées par des cordages. Ces effigies sont dotées d'un dispositif acoustique qui permet au prêtre de parler à travers la statue. Le prêtre est investi du rôle d'intermédiaire entre les forces surnaturelles et le peuple; sa machine est la statue animée. Cette ambivalence entre matérialité et surnaturel est toute entière comprise dans l'expression latine *deus ex machina*, qui signifie, littéralement, « le dieu sortant de la machine ».

Même si elles revêtent une dimension imaginaire, certaines descriptions de ces objets rituels reposent néanmoins sur des dispositifs mécaniques plausibles. Le passage anonyme suivant en témoigne : « Un jeu de cordes, de poulies et de charnières permet aux bouches de railler les terreurs du novice, aux yeux de pleurer sa mort, aux vers de le dévorer ». Comme on le verra au chapitre suivant, les masques et statues antiques jouent un rôle crucial dans le spectacle. Lors des fêtes en l'honneur de Bacchus, qui comportaient des danses, des jeux et des mystères d'initiés, et parfois qualifiées d'orgies dans leur version décadente, on suspendait une statue du dieu au-dessus de la scène, animée par une machine ou par un mécanisme.

Ainsi, Callixène (280 av. J.-C.) raconte le faste déployé par Ptolémée Philadelphie en l'honneur d'Alexandre et de Bacchus. Entre autres choses, il y avait là un char à 4 roues de 8 coudées de largeur, traîné par 60 hommes, sur lequel était assis une statuette de Mysa de 8 coudées, vêtue d'une tunique jaune brochée d'or et d'un



manteau lacédémonien. Mue par une machine, cette statue se levait sans que personne approchât la main, versait du lait d'une bouteille d'or et se rassoyait.

La réalisation la plus achevée de cette lignée d'artéfacts, c'est le *Théâtre de machine* d'Héron d'Alexandrie décrit plus haut, dont la distribution compte une vingtaine de personnages.

2.3.7. *Le jaquemart*

Le jaquemart, que l'on trouve dès 1332 à Courtrai (alors en France), est une figurine allégorique animée, placée au sommet d'un clocher ou d'un beffroi. Régulé par une horloge, le jaquemart frappe les heures sur une cloche avec son marteau. Cet ancêtre de l'automate est à la fois la représentation symbolique de l'horloger et du guetteur. Parfois, le mécanisme de commande activé par l'horloge peut être complexe et fastueux; le jaquemart devient alors un des personnages d'un véritable théâtre de machines. Vers 1340, l'horloge de l'Abbaye de Cluny, est dotée d'un véritable théâtre qui annonce les heures, à côté du calendrier perpétuel indiquant les positions, les oppositions et les conjonctions des astres ainsi que les phases de la lune. Un coq battait des ailes et chantait à deux reprises, un ange ouvrait une porte et saluait la Sainte-Vierge, le Saint-Esprit descendait sous la forme d'une colombe, le Père Éternel les bénissaient tous, et toutes les figurines rentraient alors à l'intérieur de l'horloge, au milieu d'un carillon harmonique de clochettes et de bizarres manœuvres d'animaux fantastiques, qui agitaient leurs yeux et leurs langues.

2.3.8. *Le théâtre de machines*

Entre l'âge d'or de la mécanique à Alexandrie et sa redécouverte lors de la Renaissance occidentale presque un millénaire plus tard, ont été créés, entre autres, des oiseaux artificiels animés de jets d'eau, attribués aux Arabes, qui pratiquaient et maîtrisaient l'artisanat mécanique. En Occident, il est vraisemblable que les *Miracles* et les *Moralités* montés par la Confrérie de la Passion jusqu'au milieu du 16^e siècle faisaient intervenir des vols d'anges effectués par des machines. Il est certain qu'à la fin du 16^e siècle, des machines étaient employées dans les ballets de cour pour



mettre en scène la descente de dieux sur terre. Par ailleurs, au 16^e siècle, le cheval volant est l'un des principaux thèmes mythiques où des mécaniques sont utilisées dans le cadre d'un spectacle.

Au 17^e et 18^e, les automates ont suivi deux voies distinctes mais parallèles : la *monumentalisation* et la *miniaturisation*. Sur le plan monumental, on retrouve des machines de théâtre et, dans les jardins, des jeux d'eau automatiques pour le divertissement de la cour. Les machines de théâtre produisaient les effets les plus spectaculaires (chars volants, nuages, mer agitée, etc.). Ainsi, à la fin de la première tragédie de Pierre Corneille (1606-1684), jouée au cours de la saison 1634-1635, une machine était utilisée pour mobiliser le char volant tiré par deux dragons avec lequel s'échappe Médée. Mais la grande impulsion vers la technologie nouvelle vint de l'Italie, pays du baroque, dont les créateurs ont grandement influencé l'utilisation des machines dans le théâtre.

Cette influence est manifeste dans *Ulysse dans l'Isle de Circé*, tragi-comédie de Claude Boyer, représentée au Marais en 1648, rebaptisé pour l'occasion *Théâtre des machines du Marais*. L'auteur n'est plus obligé de choisir entre une intrigue essentiellement dramatique à laquelle peuvent s'ajouter quelques rares interventions spectaculaires, ou une série d'épisodes permettant de faire descendre et remonter les divinités sans trop d'in vraisemblance. Boyer a su allier de manière assez perspicace les tensions affectives qui habitent les personnages à une application impressionnante de la technologie des machines. Le cinquième acte voit l'intrigue amoureuse au centre de la pièce résolue, et s'ensuit alors une démonstration remarquable de l'art du machiniste : Euriloche enlève Phaëtuse et s'échappe dans son navire avec Circé et Leucosie à sa poursuite dans un char volant; le Soleil fait appel à Jupiter qui foudroie le ravisseur; Phaëtuse est sauvée du navire en flammes par un dauphin qui la dépose dans le char volant.

2.3.9. Les machines à musique

Parmi les miniatures, on retrouve également des artefacts voués au divertissement, présentés et primés dans des expositions : boîtes à musique, mais aussi animaux et



androïdes. Ces dernières formes d'automates fascinaient beaucoup le public parce qu'ils donnaient l'illusion d'être vivants.

De la Renaissance jusqu'à la fin du 18^e siècle qui marque leur apogée, les automates sont actionnés par des combinaisons de rouages classiques : vis, ressort, poulie auxquels viendront s'ajouter, entre autres, le piston qui permet de constituer des chaînes cinétiques. Dans un premier temps, les machines régulées ne distinguent pas les mécanismes assurant la coordination et l'enchaînement des mouvements ou des opérations des mécanismes internes d'exécution. Du coup, chaque nouvelle séquence d'opérations exigeait un nouveau réglage des mécanismes d'exécution. Rappelons que c'est au début du 17^e siècle que le dispositif de guidage fut séparé de la structure matérielle de la machine, qui devint, ainsi, une machine à programme (section 2.1.4). Utilisée d'abord dans les métiers à tisser, la carte perforée s'est par la suite répandue au milieu du 18^e siècle, et sera utilisée pour actionner les pianos et pour les orgues de Barbarie.

L'Orgue de Barbarie a connu une grande popularité et s'est vu décliner de nombreuses façons selon les cultures. En Allemagne, cet instrument est appelé *Drehorgel*, en Italie *Organetto de Barberia*, en Hollande *Draaiorgel*, *Pierement* ou *Straatorgel*, au Danemark *Lirekasse*, en Suède *Positiv*, au Brésil et au Portugal *Realejo*, et en Espagne *Organillo*. Aux États-Unis et au Royaume-Uni, les noms varient suivant la grandeur, les caractéristiques et l'usage de l'orgue: *Street Organ*, *Barrel Organ*, *Fair Organ*, *Mechanical Street Organ* et *Band Organ*.

Au 18^e siècle, à Modène en Italie, un Barberi invente un système qui permet de jouer de l'orgue à tuyau simplement en tournant une manivelle, qui va agir sur un cylindre garni de pointes. Chaque pointe correspond à une note, et active l'ouverture de la soupape du tuyau concerné. Le cylindre est garni de milliers de pointes qui constituent le programme de la musique. Vers 1845, deux autres italiens, les frères Limonaire installés à Paris, remplacent le cylindre par une bande de cartes perforées. Les orgues des Frères Limonaire vont être équipés de grosses caisses, de triangles et d'autres percussions pour devenir ainsi les grandes orgues des manèges.



L'orgue de Barbarie est associée à la culture populaire, aux fêtes foraines. Construits artisanalement, il peut avoir des dimensions imposantes : plusieurs jeux de tuyaux sur une étendue de 27 à 60 notes. La bande de cartes perforées est pliée en accordéon. L'organiste tourne la manivelle pour faire avancer la carte sur laquelle est reproduit le morceau de musique. Chaque trou représente une note, la taille du trou représentant la longueur de la note. Les cartes sont perforées à la main. En 1990, une machine à percer les partitions pour orgues de 27 touches a été mise au point à partir d'enregistrements MIDI. L'orgue de Barbarie, issu à la fois de l'orgue portatif du Moyen-Âge, de la vielle à roue, du cylindre noté, du carillon ou de la serinette exerce une toujours une grande fascination.

2.3.10. Animaux et androïdes

Une maîtrise toujours plus grande de l'horlogerie et de la miniaturisation a permis de complexifier l'assemblage des parties mécaniques des automates. Cette complexification a débouché sur une véritable *esthétique mécaniste* : variété, fluidité et vraisemblance des mouvements. Il y a un parallèle à faire avec la situation d'aujourd'hui, où les artistes de la modélisation 3D s'emploient à créer des automates immatériels à partir de points d'ancrages décrits en coordonnées, en position. Ces automates sont recouverts d'une « texture » adaptée à leur forme, et constitués par polygones ou par d'autres stratégies géométriques. Ils interagissent avec leurs voisins ou avec leur environnement. La scène est par la suite éclairée et captée par une caméra virtuelle, et finalement rendue sur la surface de l'écran. Ce qui rapproche ces deux périodes, c'est qu'il s'y joue une même course au vraisemblable.

Le 18^e siècle marque l'apogée de l'automate en tant qu'artéfact. Les grandes expositions leur consacraient une grande place. Dans ce très grand ensemble, on a retenu deux automates (décrits plus bas), les plus connus et les plus achevés. Parmi les mécaniciens de génie, il faut compter Jacques de Vaucanson (1709-1782), à qui l'on doit à la fois le *Flûteur* et le *Canard* décrits ici, ainsi que les frères Droz, Jean-Pierre (1746-1825) et Pierre-Jacques dit Jacquet (1721-1790), à qui l'on doit, entre autres innovations, le perfectionnement de l'association tuyau sonore-piston et la



miniaturisation. Ils ont créé, entre autres, des tabatières dont le mécanisme automatique, qui représente un oiseau chantant, est réduit à quelques millimètres. Les deux exemples présentés ici illustrent chacune des deux stratégies qui furent adoptées pour réduire la distance entre la réalité et son imitation: 1) la *reproduction du mouvement* d'un être vivant (et non pas de son mécanisme intérieur) ; 2) la *reproduction* technique du *mécanisme intérieur* d'un être vivant. Nous verrons plus loin que cette quête de la similitude se double d'une quête du spectaculaire.

Le *Joueur de Flûte* ou *Flûteur* de Vaucanson (1737) est un androïde assis d'un mètre et demi, automate à forme humaine qui accomplit des gestes semblables à ceux de l'humain. À l'image d'un flûtiste, l'automate de Vaucanson jouait 12 airs différents, dont " Le Rossignol " de Blavet. Le mécanisme imite les effets et les moyens de la nature avec exactitude et perfection. Cela implique déjà une grande connaissance de la flûte traversière et de son maniement. L'automate accomplit rigoureusement les mêmes gestes et obtient les mêmes résultats : boucher les trous avec les doigts, donner le coup de langue adéquat pour produire certaines notes. Le fonctionnement du flûtiste est néanmoins totalement différent de celui du modèle humain; il ne s'agit pas d'un être biologique, mais mécanique. Par un engrenage, l'axe moteur transmet simultanément le mouvement à deux composants: la *soufflerie* génère une forte, moyenne ou faible pression, et le *cylindre à cames* constitue le programme, c'est-à-dire la pièce musicale à jouer. De gauche à droite, les tiges reliées aux différentes cames du cylindre actionnent trois doigts de la main gauche et les quatre doigts de la main droite du flûtiste. Trois autres cames permettent d'obtenir le mélange de pression désirée pour une note donnée. Les cinq dernières cames servent à placer les lèvres et à activer le coup de langue qui permet d'effectuer des forts, des crescendos et même des échos sur une anche automatique.

Le *Canard Digérateur*, chef-d'oeuvre de simulation anatomique, fut présenté par Vaucanson à l'Académie des Sciences en 1738, en même temps que *Le Joueur de Flûte*. Il est décrit dans un catalogue comme suit : « Un canard artificiel en cuivre



doré qui boit, mange, croasse, barbotte dans l'eau et fait la digestion comme un canard vivant ». Avec cet automate mécanique, Vaucanson s'attaquait à la représentation du fonctionnement organique, appelée alors *anatomie mouvante*. Le mécanisme, entraîné par un poids, comportait plus de mille pièces mobiles : bielles, comes et pistons. Il comportait même un intestin en tuyau de caoutchouc provenant du Nouveau Monde. Vaucanson fait preuve d'une rare maîtrise de son art, son analyse du dandinement du canard est très fine, vu la fluidité et le réalisme du mouvement obtenu par l'agencement des dispositifs mécaniques.

2.4. Les automates digitaux

Dérivé du calculateur analytique de Babbage, l'ordinateur est une machine automatique, disposant d'une mémoire étendue et d'une unité de commande. Cette unité de commande effectue des opérations logiques de calcul et de traitement sur des données grâce à des programmes externes. L'ordinateur n'est pas un calculateur; c'est une machine à traiter de l'information binaire à partir d'instructions elles aussi binaires ; les opérations arithmétiques peuvent être indirectement réalisées au moyen d'un programme approprié.

Les *automates digitaux* sont des programmes informatiques, c'est-à-dire des suites d'instructions qui sont compilées et exécutées par un ordinateur. Toutefois, ne considérer que le programme, c'est se confiner à l'aspect informatique de la tâche à accomplir. On néglige alors plusieurs aspects pourtant de première importance, comme le *contexte* de l'exécution de la tâche et l'*ordinateur* en tant que tel. L'automate digital recouvre donc une réalité plus large que le programme informatique.

2.4.1. Le calcul logique

Le mathématicien britannique Georges Boole (1815-1864) publie en 1854 un traité intitulé *Une investigation des lois de la pensée sur lesquelles sont fondées les théories mathématiques de la logique*. Dans ce traité, Boole voulait 1) étudier les lois fondamentales des opérations intellectuelles du raisonnement, 2) exprimer ces



lois dans le langage du calcul 3) définir la science de la logique à partir de ces bases et 4) construire ses propres méthodes.

Boole définit la logique comme un calcul de signes arbitraires. Le symbolisme utilisé est celui de l'algèbre, qui permet de se détacher des quantités pour exprimer la valeur de vérité. Le calcul logique est un ensemble d'opérations algébriques sur les symboles représentant des classes. Les signes $+$, $-$, X et $=$ représentent respectivement la *réunion* ou la *somme* de deux classes, l'*exception* de certains éléments dans une classe donnée, le *produit logique*, et l'*identité extensionnelle*, c'est-à-dire l'identité des membres composant respectivement chacune des deux classes. Boole propose des lois fondamentales qui constituent, en fait, des tables de vérité.

L'impact de l'algèbre de Boole pour l'ordinateur est majeur. On lui doit les opérations logiques de base effectuées sur les codes binaires par le processeur : la conjonction, la disjonction et la négation. La pensée algébrique est le résultat de plusieurs siècles de développement de la pensée symbolique. La séparation de plus en plus nette entre les *mécanismes* de la pensée, les opérations cognitives, et le *contenu* de cette pensée constitue l'un des principaux développements de la pensée symbolique. Cette séparation constitue un processus d'abstraction, car pour isoler les mécanismes de la pensée de son contenu, on réduit toute référence à une quantité; la sémantique, le sens qu'on accorde aux choses, se trouve ainsi mis de côté.

Charles Sanders Peirce (1839-1914) est le philosophe et logicien américain à qui l'on doit, en grande partie, les fondements de la sémiotique. Il a également beaucoup contribué au domaine du *calcul des relations*, en concevant un système de signes graphiques et de diagrammes qui permet de représenter le raisonnement et de l'intégrer aux opérations des machines (1867).

La thèse de Claude E. Shannon, soutenue en 1937, est intitulée *De l'analyse symbolique des relais et des circuits de communication*. Shannon y propose une première mise en application aux réseaux électriques des règles de l'algèbre logique proposée par Boole. Il relève une similitude entre la logique des circuits et



les opérateurs de la logique formelle, ce qui lui permet d'envisager une implémentation physique du raisonnement déductif de l'algèbre des propositions à partir des tables de vérité. Sa réflexion est poussée; il généralise l'analogie proposée en énonçant que l'on peut automatiser au moyen de relais toute opération intellectuelle pouvant se définir par un nombre déterminé d'étapes mettant en jeu les fonctions de base : l'union, l'intersection, la négation, l'implication et la comparaison. Il suggère que de tels circuits sont capables de prendre des décisions, et de faire varier la procédure en fonction des résultats intermédiaires de ses calculs logiques. Les impulsions représentant les chiffres binaires à traiter seront envoyées dans les circuits appropriés. Chacune des impulsions ouvre ou ferme un ou plusieurs relais d'un circuit de commutation, organisé pour effectuer une des fonctions de base. À la sortie du circuit, les impulsions finales représentent, en binaire, le résultat de l'opération commandée.

2.4.2. La machine de Turing

Alan Mathison Turing (1912-1954) présente une thèse intitulée *Sur les nombres calculables et leur application au problème de la décidabilité* en 1936. Turing caressait le projet de mécaniser l'intelligence, ce qui en fait un des précurseurs de l'intelligence artificielle. Pour lui, calculer, c'est penser, et la faculté de raisonnement humain est assimilée à la capacité à calculer des machines. Il propose une théorie de l'automate algorithmique universel, qui sera par la suite désignée par l'expression « machine de Turing ». Il démontre que toute fonction humainement calculable est aussi calculable par une machine de Turing et réciproquement. La section suivante est consacrée à cette machine théorique.

Cet automate abstrait constitue une mémoire universelle à la fois pour les instructions et les données. Son fonctionnement est discontinu. Passant d'un état à un autre, il effectue des séquences enchaînées d'opérations élémentaires, étape par étape. Ces opérations sont de deux types:

- 1) le traitement proprement dit, qui exécute les instructions correspondant à un problème particulier



- 2) la supervision du déroulement du traitement, la modification des adresses, la prise de décisions, etc.

Une machine de Turing comporte

- 1) un répertoire fini de symboles ;
- 2) une bande potentiellement infinie divisée en cases successives ; certaines d'entre elles peuvent rester vides, mais chacune ne peut contenir qu'un symbole à la fois ;
- 3) un dispositif d'effaçage qui permet de supprimer un symbole figurant dans une case ;
- 4) un dispositif de lecture et d'écriture adapté à la bande, qui peut lire une seule case à la fois, et peut écrire sur une case vierge uniquement ;
- 5) un dispositif d'entraînement qui déplace la bande d'une case à la fois en avant ou en arrière ; la bande peut aussi rester immobile ;
- 6) une table de comportements qui, à chaque étape du fonctionnement de l'algorithme, définit une situation ;
- 7) une unité de commande qui permet de mettre la machine dans les divers états définis dans la table de comportement.

Ce formalisme permet de produire des machines capables de modifier d'elles-mêmes leur état interne. La capacité de ces machines est fonction du nombre d'états ainsi que de la richesse et de la quantité de symboles. Toutefois, la contrepartie de la puissance et de la généralité de cette machine abstraite, c'est que beaucoup de problèmes sont indécidables. En imposant plus de contraintes aux machines, on diminue leur puissance d'expression, mais on augmente leur pouvoir décisionnel.

L'automate à états finis, contribution de Alan Turing, permet une représentation abstraite du traitement requis ainsi que des données. Sur la base de cette représentation algorithmique, des programmes ont pu être développés à partir de langages de programmation, non seulement pour effectuer les traitements, mais aussi pour communiquer.



2.4.3. L'ordinateur

Le premier automate digital est le EDVAC — *Electronic Discrete Variable Automatic Computer*. Sa construction a débuté en 1945 et s'est terminée en 1951 sous la direction de John von Neumann (1903-1957). Celui-ci a voulu concevoir l'architecture de cet automate à l'image du cerveau humain, et ainsi simuler le fonctionnement de l'intelligence humaine à l'extérieur du cerveau humain.

Sa conception de l'intelligence humaine est *atomiste*. Pour lui, les capacités intelligentes du cerveau sont le fruit de la combinaison d'éléments primaires simples. Cette méthode de compréhension par décomposition avait déjà été utilisée avec succès dans plusieurs domaines: en chimie avec le tableau périodique, en physique avec l'atome, en biologie avec la molécule et la cellule. De même, la complexité de la pensée se réduit à une combinatoire de neurones. Au moment de concevoir les plans d'une machine, von Neumann recherche l'unité de base du cerveau, l'équivalent matériel de l'atome. Il établit une comparaison entre le neurone du cerveau humain et le tube à vide dans un circuit binaire. La mémoire est conçue comme le siège du raisonnement, le lieu où les mouvements de l'information produisent la réflexion consciente. Les impulsions électroniques auxquelles on a préalablement donné une signification sont les pensées.

Sa représentation du fonctionnement du cerveau est analogue à celle de la machine à changements d'états de Turing. L'ordinateur est donc, fondamentalement, un automate à états finis ou *discrets*, c'est-à-dire complètement distincts les uns des autres, qui effectue des opérations logiques sur des informations transcodées en binaire, les unes à la suite des autres.

C'est ainsi que l'ordinateur est *séquentiel*: de par sa structure, il est incapable d'effectuer des opérations en parallèle, ce qui, pourtant, conviendrait mieux aux données structurées en matrices ou en réseaux. Au cœur de l'ordinateur, une horloge électronique synchronise les opérations internes en cycles, et réactive plusieurs milliers de fois à la seconde chacune des impulsions contenues dans les circuits de la machine.



Comment ne pas voir le lien avec le rôle de l'horloge comme principe de régulation des machines analogiques ? Concluons-en provisoirement que les machines digitales et les machines analogiques, malgré les différences de nature de leurs composants (instructionnels d'un côté et matériels de l'autre) ont beaucoup plus en commun qu'il n'y paraît à première vue.

L'ordinateur est une machine à programme entièrement électronique, et les programmes sont enregistrés dans la mémoire interne. Auparavant, dans les premiers calculateurs automatiques, la mémoire interne n'était utilisée que pour stocker les résultats intermédiaires des calculs. Le système de programmation et le stockage des instructions de l'automate étaient externes, et donc distincts de la mémoire interne. Du coup, aucun événement survenant pendant le traitement ne pouvait modifier l'exécution de la séquence d'instructions prévue dans le programme.

L'ordinateur pourrait être qualifié de méta-automate dans la mesure où, d'une part, il peut engendrer une infinité d'automates particuliers en fonction des programmes exécutés, et que, d'autre part, il constitue lui-même un automate, constitué d'un circuits de plus en plus intégrés et de plus en plus miniaturisés. Physiquement, ce méta-automate est composé d'un processeur, d'une mémoire, d'une horloge et de périphériques lui permettant d'échanger des informations avec le monde extérieur. Neutre, cette architecture permet de faire fonctionner des automates formels de toutes sortes. Ces automates sont dits formels parce qu'ils ne sont dotés d'aucune autre matérialité que celle d'une représentation au moyen d'un langage de programmation, a) de la séquence d'instructions qui constitue le programme de transformation de l'information et b) des données, représentation digitale de la réalité échantillonnée.

En d'autres mots, les automates digitaux peuvent appliquer toutes sortes de transformations aux données les plus diverses. Deux conditions doivent être réunies pour qu'ils puissent opérer ces transformations. Premièrement, il faut que les données puissent être représentées sous forme de symboles non-ambigus, appartenant à des suites finies, exprimables sous forme de code numérique et



insérés dans une structure de données appropriée (cf. 1.5.6). Deuxièmement, il faut que les transformations soient systématisées en procédures régies par les règles de la logique symbolique, exprimées dans le formalisme d'un langage de programmation (cf. 1.5.5). L'ordinateur se définit donc une machine électronique à programme numérique (cf. 1.5.2) qui sert au traitement de l'information (cf. 1.3.4).

2.4.4. Les variables

On a déjà abordé en détails la programmation procédurale qui permet d'exprimer des commandes à un ordinateur dans un langage formel ou mathématique (cf. 1.5.5). Les programmes sont traduits en programmes-objet dans un langage machine spécifique à un compilateur ; ces programmes conservés dans des fichiers binaires peuvent être exécutés à volonté. D'autres programmes peuvent être exécutés directement par un interpréteur. Contrairement aux instructions en code binaire, de tels programmes sont lisibles, ce qui en facilite l'entretien. De plus, ils sont à peu près directement transférables d'un type de machine à un autre, à condition, néanmoins, d'utiliser les compilateurs appropriés ; en outre, ils n'exigent pas une connaissance approfondie de l'architecture et du fonctionnement de l'ordinateur.

La séparation qui s'est opérée graduellement entre la matérialité des machines et le programmeur a contribué à standardiser les principaux instruments de décision et de traitement qu'offrent les différents langages: IF THEN ELSE, FOR, WHILE, DO, etc. (alors que ces langages ont chacun une finalité et un design différent). Cette séparation a aussi contribué à normaliser et à uniformiser les opérations et les actions. L'apparition de la *variable*, notion empruntée à l'algèbre, a permis au programmeur de se dégager de la gestion directe des espaces mémoires. Grâce aux variables, les langages de programmation permettent *d'assigner un nom* aux emplacements de mémoires requis pour conserver les données durant le traitement. En utilisant des étiquettes symboliques plutôt que des adresses numériques absolues pour conserver les informations, on peut manipuler des valeurs sans avoir à se soucier de leur emplacement effectif en mémoire. Les variables permettent de généraliser les procédures en se voyant associer des valeurs en contexte. Une



variable ne peut contenir qu'une valeur à un instant donné, mais cette valeur pourra évoluer sous l'action des instructions du programme.

La valeur des variables peut être de différents types : de type numérique, de type logique (vrai ou faux), ou de type " chaîne de caractères " (elle consiste alors en une suite de caractères délimitée par des séparateurs, habituellement le guillemet). À chacun de ces trois types correspond certaines opérations. Les valeurs numériques font l'objet d'opérations mathématiques, alors que les chaînes de caractère font l'objet de concaténations, d'extractions, de substitutions, etc. Des langages de programmation ont précisément été consacrés aux chaînes de caractères; PERL en est un exemple connu.

La nomenclature utilisée pour désigner les variables est capitale. Leur contenu, leur usage et leur type doivent être exprimés dans une représentation la plus naturelle et la plus appropriée possible vu les objectifs du programme. Mais en même temps, les caractères disponibles sont limités, et des noms trop longs diminuent la lisibilité des programmes. Souvent, il faut utiliser une abréviation comme symbole pour la variable, et en commentaire, fournir la notation complète sous une forme la plus naturelle possible.

2.4.5. Les listes

Les listes — traduction française de l'anglais *array* —, c'est ce qui permet de représenter des données non numériques de façon « naturelle », et de les ordonner de manière à en représenter également la structure. Lorsqu'elles sont affichées, les listes apparaissent entourées de parenthèses ou de crochets carrés, et les items de la séquence ordonnée qui la constitue sont séparés par des virgules. Les items de la liste peuvent être des *constantes* (nombres, chaînes de caractères) ou des *variables*. De plus, les items peuvent eux-mêmes constituer des listes. C'est ainsi qu'un tableau sera représenté par une liste, dont chacun des items est lui-même une liste représentant les items d'une rangée du tableau.

Les listes permettent l'accès indexé aux items. Ainsi, le temps requis pour retrouver le dernier item de la liste est le même que celui qui est nécessaire à retrouver le



premier. Toutes les opérations qui s'appliquent à des variables peuvent aussi s'appliquer aux items d'une liste. Certaines instructions permettent d'ajouter, de modifier et de retirer les items d'une liste. Les listes peuvent être utilisées non seulement pour repérer les données qui sont conservées à une position donnée, mais aussi pour contrôler l'application itérative d'une procédure à chacun des items d'une liste (*foreach*).

Il existe d'autres structures de données, plus complexes et plus puissantes. La *liste associative* est une liste qui a toutes les caractéristiques qu'on vient de mentionner, la différence étant que ses items sont composés de deux parties, une *clé* et la *valeur* proprement dite. Cette fois, l'index qui permet d'accéder aux items n'est pas numérique (relié à la position de l'item), mais symbolique. Avec la clé, l'accès à une valeur est néanmoins aussi rapide qu'avec l'adresse. Ces listes sont parfois appelées *listes à propriétés*, parce qu'elles permettent de regrouper les caractéristiques des entités à traiter. Ce genre de représentation sera utilisée, entre autres, pour exprimer des connaissances destinées à la prise de décision d'un système expert ou à faire l'objet d'autres traitements « intelligents ». Ce genre de traitements est au centre de la section suivante.

2.4.6. *Le principe de décomposition*

La programmation pourrait être définie comme un art du langage. Un de ses principes les plus importants, c'est le *principe de décomposition*. Ce principe joue sur tous les plans. Au plan du programme lui-même, il y a à la fois une séparation et une décomposition entre les instructions et les données qui sont traitées.

Pour résoudre un problème à l'aide d'un ordinateur, il faut passer par trois phases. Premièrement, pour *représenter* les aspects critiques du problème à résoudre, il faut *créer* des objets formels compréhensibles par un ordinateur, suivant un schème régulier, la structure de données. Deuxièmement, il faut *computer* la séquence des instructions sur les données afin de produire un résultat attendu. Troisièmement, il faut *interpréter* les résultats. Autrement dit, il faut traduire les résultats obtenus lors



de la computation en une information signifiante sur la situation et sur la solution du problème.

Lorsque le problème est de nature arithmétique, sa représentation est transparente. Le stockage et la manipulation des nombres, une fois que ceux-ci sont convertis dans leur représentation binaire, se font directement dans les registres de l'ordinateur. Par contre, lorsque le problème à résoudre à l'aide de l'ordinateur n'est pas de nature arithmétique, c'est-à-dire dans la grande majorité des cas, le mode de représentation choisi et son utilisation sont nettement moins transparents. Les données doivent représenter les aspects significatifs du problème non seulement dans une forme intuitive, naturelle, facile à lire, mais aussi à travers une structure formelle qui soit suffisamment uniforme pour permettre des manipulations standardisées. Si les données et les procédures sont insuffisamment structurées, les instructions redondantes prolifèrent, ce qui nuit au fonctionnement de l'automate, à sa maintenance, à son évolution et à sa réutilisation.

2.4.7. Les fonctions

Il faut séparer, parmi les instructions, les *instructions de base* et les *instructions de structuration*. Les *instructions de base* manipulent les variables et exécutent certaines opérations sur leur valeur. Les *instructions de structuration* du programme déterminent la séquence des instructions de base, en administrant les tables de décisions, en régulant la répétition, etc. Parmi les instructions de base, il faut distinguer les énoncés déclaratifs et les énoncés impératifs. Les *énoncés déclaratifs* fournissent l'information nécessaire à la conduite des opérations; par exemple, ils donnent le nom des variables, spécifient leur type ou tout autre contrainte sur la valeur, la longueur et le type des listes, etc. Les *énoncés impératifs*, quant à eux, spécifient les opérations de traitement, d'entrée et de sortie des données, etc. Tout au long de l'exécution du programme, les variables, listes, tableaux, fichiers déclarés peuvent être utilisés. L'accès à leur valeur (ou à la valeur d'une de leurs propriétés dans le cas d'une structure de données) s'opère à partir des identificateurs symboliques déclarés.



Pour décomposer, au niveau des instructions, l'expression individuelle de chacune des occurrences et de ses constituants particuliers en variables, il faut distinguer entre nouveauté et redondance. Ainsi, au lieu de ramener toutes les tâches à un très long enchaînement d'opérations élémentaires, il faut *simuler le processus* par des fonctions appliquées à des paramètres (les variables), sur lesquels on peut agir sans passer par la continuité étalée des états successifs et redondants. Cette opération par laquelle on passe d'une suite étalée à un ensemble de fonctions s'appelle la *fonctorisation*.

Une fonction, ou sous-routine, accomplit une séquence d'opérations donnée à l'intérieur d'un processus. Les suites d'instructions qui interviennent dans le processus à plusieurs reprises moyennant quelques variations mineures et régulières doivent ainsi être regroupées en fonctions. Une fois la fonction déclarée et les paramètres spécifiés, celle-ci peut être appelée à partir d'une autre fonction, avec des arguments spécifiques à la situation. Les *arguments* d'une fonction sont les valeurs passées en paramètre lors de son appel, qui serviront de valeur aux variables impliquées dans les instructions. Une fonction retourne une valeur qui peut soit servir d'argument à une autre fonction, soit être assignée à une variable, soit faire l'objet d'une comparaison dans un branchement conditionnel, etc. Les fonctions, qu'elles soient simples ou complexes, sont toujours décomposables.

Plusieurs critères permettent de juger si le niveau de décomposition en une sous-routine distincte est pertinent. Premièrement, elle est le fruit d'une bonne *généralisation*. Cependant, plus la routine est générale, plus le nombre de paramètres requis pour tenir compte de la variété des contextes d'application est grand; par conséquent, il en va de même pour les branchements conditionnels. Deuxièmement, elle est *optimisée* lorsque les stratégies de codage utilisées se révèlent les plus efficaces en termes de temps de calcul, de quantité d'accès mémoire et du volume occupé par les résultats intermédiaires. Troisièmement, la sous-routine est *compacte* lorsque le nombre d'instructions est le plus réduit possible. Pour ce faire, on a souvent recours à l'itération, à la répétition d'un bloc d'instructions (avec possibilité d'interruption), ou encore à la récursivité.



Quatrièmement, la documentation doit spécifier clairement le traitement que suit la sous-routine, le type et les contraintes des valeurs que peuvent prendre les différents arguments déclarés, et le résultat ou les messages d'erreur qui sont retournés.

Une *procédure récursive*, c'est une procédure qui s'utilise elle-même comme sous-procédure. Ce genre de procédure est particulièrement bien adaptée au traitement d'expressions symboliques structurées en arborescence ou en une autre forme de réseau moins contrainte. La structure du programme correspond alors à la structure de données. Toutefois, la récursivité pose un problème : les variables dans le programme correspondent à des emplacements précis dans la mémoire, et lorsque le programme est appelé par lui-même, les mêmes emplacements seront utilisés, remplaçant ainsi le contenu précédent. Les langages de programmation qui supportent la récursivité utilisent une liste dans laquelle une nouvelle cellule est ajoutée à chacun des appels récursifs d'une sous-routine donnée. Cette liste fonctionne comme une pile: toutes les insertions, toutes les suppressions et, d'une manière plus générale, tous les accès se font au début de la liste. Ainsi, lorsqu'une sous-routine s'achève et retourne son résultat, le contenu des registres sauvegardés dans la pile sont restaurés. LISP — *LIS*t *Processing* —, programme développé au Massachusetts Institute of Technology (MIT) à la fin des années 1950, est l'un des premiers langages de programmation à tirer pleinement avantage de la computation récursive. Comme on le verra dans la section suivante, ce langage a été utilisé dans de nombreuses implantations informatiques d'intelligence artificielle.

2.4.8. Modularisation

La construction de systèmes complexes nécessite une *modularisation*. La modularisation, c'est un découpage qui permet la constitution de modules, ou morceaux de code, délibérément distincts les uns des autres. Cette opération rappelle la décomposition d'un processus en sous-routines (cf. *supra*), mais la modularisation n'a pas la même finalité. Elle vise à limiter le domaine d'intervention des programmes qui seront assemblés pour la réalisation d'une tâche donnée. Au lieu de devoir toujours tout recommencer à partir des instructions de base, on peut,



en intégrant des modules déjà optimisés, profiter de certains acquis passés. De cette manière, les systèmes peuvent atteindre des niveaux de complexité de plus en plus grand, tout en se voyant affecter des ressources limitées pour leur développement. Pour développer un module, il faut, par rapport, aux autres modules, 1) isoler et identifier les canaux de communication des données et des variables de contrôle ; 2) définir un protocole de structuration de données et 3) statuer sur la teneur des échanges de données.

L'assemblage de modules en chaînes de traitement est pratiquée à large échelle par les firmes de consultants en système d'information (cf. 1.6.2) qui se définissent comme des *intégrateurs*. Linus Torvalds, né en 1969 à Helsinki, Finlande, est un étudiant programmeur qui, à l'âge de 21 ans, a écrit un système d'exploitation de l'ordinateur entièrement libre de droits et à code public, appelé LINUX, qui est fondé sur l'assemblage de modules fournis par la communauté. LINUX constitue actuellement la seule alternative au monopole des grands fabricants de logiciels.

Le principe de décomposition permet le passage du raisonnement individuel au raisonnement collectif. Au lieu d'être conçu pour résoudre un problème donné, le programme est développé pour une classe de problèmes apparentés. Cette approche de la programmation constitue un saut semblable au passage de l'arithmétique ordinaire à l'algèbre des grandeurs dans la construction des automates. Ainsi, de la même façon qu'on est passé de la présentation de nombres spécifiques à la notation d'inconnues par des symboles, on est passé d'adresses mémoires occupées par des données spécifiques aux variables qui jouent le rôle de constantes indéterminées. Une autre étape importante est celle qui a consisté à passer d'une suite déployée et redondante de toutes les instructions nécessaires à l'accomplissement d'une opération à des séquences de sous-routines avec des paramètres instanciés en contexte d'exécution. A chacun de ces passages, un niveau d'abstraction supplémentaire est franchi.

2.4.9. *L'approche orientée objet : une société d'automates*



Nous venons de voir que le principe de décomposition repose sur plusieurs abstractions: les *structures de données*, principalement en listes, les fonctions ou *sous-routines* comme structuration des instructions, ou encore les *modules* qui constituent des assemblages de fonctions servant à effectuer des tâches données.

En 1976 est proposée une approche intégrée à la décomposition complètement inédite : la *programmation orientée objet*, avec le langage Smalltalk. Ce langage implémente les concepts *d'encapsulation des données* et *d'héritage des propriétés* (l'héritage désignant la transmission de la structure d'un modèle parent à des enfants ou " instances " du modèle). L'approche par les objets est une approche intégrée. Les objets comportent 1) un *identificateur* unique stocké dans une variable ou une liste 2) une *partie structure de données*, qui permet de déclarer, pour les objets de sa classe, la valeur de l'ensemble des propriétés pertinentes en regard du traitement à accomplir, 3) une *partie procédurale*, les méthodes, qui sont des fonctions ou sous-routines instanciant les caractéristiques exposées plus haut, mais qui se voient déclenchées à la réception d'un message par l'objet et 4) une *documentation* qui spécifie, en termes de paramètres, les interactions que l'objet peut avoir avec son environnement. C'est ainsi que l'on peut définir un objet comme une entité computationnelle cohérente, définie par des comportements et une structure en propriétés. Un programme devient un ensemble d'objets en relation les uns aux autres et avec l'environnement. Ensemble, ces objets constituent non seulement une population, mais une société, un système de classes composés d'individus.

Le seul moyen de communication entre les objets est l'envoi de messages. Pour appeler une méthode, un message est envoyé pour exécuter une action donnée, avec telle et telle donnée comme argument. Le message sera composé soit d'une requête, pour laquelle l'objet expéditeur attend une réponse, soit d'une transmission univoque, qui n'attend aucune réponse. Un message est une structure formelle <objet destinataire, méthode à exécuter, arguments>. Les arguments sont les valeurs passées en paramètre à la méthode activée.



Il existe deux grands types de méthodes : les *fonctions* et les *services*. Le terme *fonction*, qui renvoie à la programmation procédurale, désigne ici une activité menant à l'obtention d'un résultat ou à la satisfaction d'un besoin. Les fonctions sont étroitement liées aux propriétés, car l'action des fonctions peut dépendre des valeurs des propriétés, ou bien venir modifier celles-ci. Un *service* est un comportement spécifique qu'adopte un objet à la demande d'un autre objet. Les services permettent d'établir des principes de spécialisation et de coopération entre les objets. Un objet peut ainsi sous-traiter une partie de son activité en demandant certains services à d'autres objets. Une fois déclaré, l'objet est en attente permanente de messages. À la réception d'un message, l'objet devient actif et effectue la méthode spécifiée, manipule son espace de données, ou envoie un message à d'autres objets. Ainsi, la programmation est dirigée par les données.

Les objets comme tels ne sont pas déclarés. En fait, c'est pour une *classe* d'objets que se fait la déclaration des propriétés, des valeurs par défaut des objets, et des méthodes qui définissent leur comportement. Le concept de classes fut proposé dès 1967 par le langage Simula. Les objets sont *instanciés* à partir d'une classe. Pour chaque classe d'objets, il existe une méthode permettant de construire des individus qui se verront associer un identificateur unique, dont les propriétés « héritées » (de la classe) auront des valeurs particulières. Ces valeurs peuvent être passées en argument à la méthode de création. Par exemple, une classe d'objets nommée *boîte* peut avoir trois propriétés : [boîte (#format, #couleur, #prix)], le message <boîte, création, moyen, blanche, 2.25> entraîne la création d'un objet singulier [boîte_à_cadeau (#format = moyen, #couleur = blanche, #prix = 2.25)].

À l'*exécution*, il n'existe que des objets; mais lors de la *programmation*, on définit des classes. L'instanciation des objets fait partie de l'exécution. Deux instances d'une même classe se distinguent par leur identificateur et par la valeur de leur propriétés. Si la valeur des propriétés n'est pas spécifiée par des arguments lors de la création de l'instance, la valeur de la classe sera attribuée « par défaut » à cette instance. Le procédé par lequel une classe dérivée reçoit les propriétés d'une classe de base est appelé *héritage* ; ce processus est transitif. L'héritage est dit *simple* si



une classe ne peut directement hériter que d'une superclasse. Dans ce cas, le graphe d'héritage est un *arbre*, et un parcours dans la hiérarchie est linéaire. L'héritage est dit *multiple* quand une classe peut directement hériter de plusieurs classes de base. Dans ce cas, le graphe d'héritage est un *réseau*. Cela diminue les contraintes dans la transmission lorsqu'on veut générer des objets dont la structure se constitue en assemblant les propriétés héritées de plusieurs parents. En contrepartie, l'héritage multiple peut aussi engendrer des conflits: une même propriété peut se voir héritée avec des valeurs différentes.

La diffusion de l'approche orientée objet a augmenté le nombre et le niveau d'abstraction des composantes réutilisables. La réutilisation favorise l'extensibilité: une classe peut être dérivée d'une autre et ainsi en étendre les capacités, les spécialiser, les adapter, etc. Cette méthode facilite la modélisation: elle permet de représenter plus facilement les relations entre les composantes, les objets, les tâches à accomplir et les tables de décision.

La *modélisation objet*, c'est la décomposition d'un phénomène 1) en événements élémentaires (en identifiant la nouveauté et la redondance) et 2) en objets qui constituent les actants des événements. Pour les événements, on procède à l'identification de circonstances — les fréquences, les corrélations, les liens de cause à effet, les conséquences, etc. Pour les objets, on procède à l'inventaire des quantités et des qualités pertinentes.

2.5. L'intelligence artificielle

L'objectif que poursuivaient les trois grandes figures de l'histoire de l'ordinateur — Turing, Shannon et Von Neumann — n'était pas de nature instrumentale. Leur quête, c'était la reproduction du cerveau humain, voire du raisonnement, à partir des lois de la pensée. Cette idée que l'intelligence pouvait être simulée dans une machine était largement répandue dans le milieu des sciences et des techniques liées au développement de l'ordinateur.

En 1956, le *Darmouth summer research projet on artificial intelligence* inaugure un champ de recherche majeur, qui a mobilisé des ressources humaines et matérielles



gigantesques et aux retombées incalculables, même si l'intelligence s'avère hors de la portée des machines. Ces retombées marquent tous les aspects de notre vie : certains des principes à la base de l'intelligence artificielle sont intégrés aux grands systèmes informatiques administratifs, techniques et scientifiques ; on les retrouve aussi à l'oeuvre dans les jeux.

2.5.1. Connexionnisme versus rationalisme

Deux approches fondamentalement opposées ont caractérisé le développement de l'intelligence artificielle : l'*approche connexionniste* et l'*approche rationaliste*. Dans les deux cas, l'intelligence se définit de manière *atomiste*, dans le sens où elle se déconstruit en primitifs, ou éléments de base: neurones pour les uns, symboles pour les autres. Ainsi, est réputé intelligent tout dispositif dont les éléments sont dotés d'un seuil minimal de complexité, que ce dispositif soit naturel ou artificiel. A cette époque, on pensait que toutes les opérations de la pensée pouvaient être décrites par l'algèbre de Boole, et, du coup, que les propriétés du cerveau pouvaient être décrites par l'algèbre logique.

Suivant l'approche traditionnelle et aussi la plus explorée de l'intelligence artificielle, qu'on peut appeler *approche du haut vers le bas*, on crée de l'intelligence artificielle non pas en reproduisant l'organisation de la structure physique du cerveau, mais en répliquant son comportement principal, à savoir le raisonnement, à l'aide de programmes d'ordinateur. Ces recherches s'inscrivent en droite ligne de la tradition du rationalisme et de l'empirisme logique qui remonte à Platon. L'intelligence est alors définie comme la capacité d'une structure physique à traiter des symboles. Au plan de leur fonctionnement, cerveau et machine sont mis sur le même pied: ils sont tous deux vus comme des systèmes physiques de traitement symbolique, ce qui présuppose que l'on sépare l'être informationnel de ses supports matériels.

Cette vision de l'intelligence est sous-tendue par une idée de base, à savoir que notre cerveau n'a pas directement accès au monde. Il fonctionne plutôt sur une représentation interne de la réalité, représentation constituée d'un ensemble de structures de symboles. Cette représentation interne des structures de symboles



peuvent prendre la forme de n'importe quel pattern physique : tableau de transistors dans un ordinateur digital, filaments de neurones excités dans un cerveau biologique. Un système intelligent, cerveau ou machine, peut manipuler des structures symboliques pour les transformer, les étendre, les diviser, les modifier en détruisant certaines d'entre elles et en en créant de nouvelles.

Suivant une autre approche de l'intelligence artificielle, qu'on peut appeler *approche du bas vers le haut*, l'intelligence s'atteint en construisant une réplique électronique du cerveau humain, et donc en reproduisant les types de connexions physiques trouvées dans le cerveau de l'animal et de l'humain. Selon ce point de vue, c'est la complexité du réseau de neurones en lui-même qui fait les caractéristiques de l'intelligence humaine. L'ordinateur séquentiel tel que le concevait von Neumann est jugé inadéquat, et se voit préférer une architecture de traitement massivement parallèle.

2.5.2. Test de Turing

En 1950, Alan Turing propose un test pour vérifier l'intelligence d'une machine. Une personne pose des questions à la fois à une machine et à une personne via un télétype durant une période de temps plus ou moins longue (au moins une heure). Suivant ce test, un système peut être qualifié d'intelligent si un observateur évalue que le comportement de la machine et de l'humain sont identiques sur un plan fondamental, même s'ils sont différents en apparence. La plupart des commentateurs s'entendent pour critiquer ce test, qui relève d'une vision très behavioriste de l'intelligence. Largement répandu en son temps, le behaviorisme ne considère que les comportements sans se préoccuper de leur cause psychologique. Même si la conversation n'est sans doute pas le meilleur moyen de juger l'intelligence, le test de Turing a l'intérêt de poser la question de savoir ce qu'est l'intelligence, comment l'observer et comment la qualifier. En interpellant la communauté des psychologues et des philosophes, les scientifiques investis dans ce projet d'intelligence artificielle furent à la source d'avancées inestimables dans la compréhension de l'intelligence humaine en elle-même. Ce questionnement a même fait émerger de nouvelles disciplines : les *neurosciences*, qui étudient les réseaux



neuronaux, et les *sciences cognitives*, consacrées à l'étude du traitement symbolique.

2.5.3. *L'intelligence en question*

La définition que chacun se forge de l'intelligence est au centre du débat sur l'intelligence artificielle. Suivant une définition fonctionnelle et pragmatique, l'intelligence se réduit à la capacité à utiliser et à manipuler des systèmes symboliques comme ceux que l'on retrouve en mathématique et en logique. Sont alors considérées comme intelligentes les machines qui résolvent des problèmes dans des situations pratiques, prennent des décisions et exhibent un niveau de performance acceptable dans un rapport coût-bénéfice optimum. Toutefois, assimiler tous les modes de pensée au calcul et au raisonnement logique est pour le moins réducteur. Le type de raisonnement que caractérisent les chaînes d'inférences classiques du type « si A implique B et que A est vrai, B est vrai » est tout à fait à l'opposé de la pensée créative qui, au delà des calculs de vérité sur des faits, implique la totalité de la personnalité, et donc non seulement ses connaissances, mais aussi ses sentiments, ses croyances, ses valeurs, ses pulsions, son intuition, son imagination, bref, tout qui constitue la richesse et la diversité de l'expérience.

Une série de critères déterminant le niveau d'intelligence d'un système se sont vus définis. Un large consensus s'est établi autour des critères ci-dessous. Pour être qualifié d'intelligent, un système doit être capable

- a) de prendre en compte des situations qu'il n'a pas encore rencontrées
- b) de fonctionner efficacement lorsque les informations sont imprécises, erronées ou incomplètes
- c) de tirer des leçons du passé
- d) de prendre des initiatives
- e) d'accéder sélectivement à de grandes quantités de connaissances
- f) de planifier des scénarios d'actions et de les accomplir
- g) de transgresser les règles lorsque rien d'autre ne semble fonctionner.



2.5.4. Utopie motivante

Est-il possible de créer une machine pensante? Cette question au centre du débat sur l'intelligence artificielle est facile à trancher: ce n'est pas demain la veille que l'on pourra écrire des programmes pour développer une conscience, un psychisme, etc. On a tenté de justifier cette croyance dans l'avènement d'une forme d'intelligence artificielle de plusieurs manières: l'utopie motivatrice, le refoulement, le technocentrisme et même la misogynie.

L'*utopie motivatrice* (ou motivante) renvoie au besoin qu'éprouve l'humain d'attribuer un sens plus grand que nature à ses gestes, pour ainsi se donner de l'aplomb, du courage, de l'énergie, etc. Par ailleurs, la primauté déclarée et reconnue de la technologie réflexive et scientifique sur tous les autres aspects de la vie provoque un *refoulement* de l'imaginaire, ce qui fait réapparaître régulièrement certains fantasmes, qui permettent de rééquilibrer les affects. Il ne faut pas oublier que dans le for intérieur des individus, une relation se noue entre le technicien — côté droit du cerveau — et l'artiste — côté gauche du cerveau. Le *technocentrisme*, c'est la tendance à privilégier son domaine d'expertise technique au point d'en faire le seul modèle de référence qui permette de comprendre le monde. Enfin, la *misogynie*, c'est la haine et le mépris pour les femmes. Le milieu de l'intelligence artificielle est en très grande majorité un monde d'hommes. Le rêve de l'intelligence artificielle, n'est-ce pas aussi le rêve de reproduire l'homme sans l'intermédiaire d'une femme ?

L'histoire de l'intelligence artificielle est longue, tumultueuse et variée. On va s'attarder maintenant sur les fondements de cette histoire, et ce en partant de quelques jalons importants de l'histoire de l'automatisation du raisonnement : les réseaux neuronaux, les planificateurs, les systèmes experts et les agents.

2.5.5. Les réseaux neuronaux

Dès 1943, les mathématiciens W. Mc Culloh et W. Pitts démontrent qu'un réseau neuronal est capable d'effectuer des opérations logiques, suivant les principes de Boole. En 1949, D. Hebb démontre comment un réseau neuronal peut non



seulement traiter l'information, mais aussi *apprendre*. Les réseaux neuronaux constituent un type particulier d'automates digitaux, dont l'architecture se fonde sur une modélisation du cerveau des mammifères. Un réseau neuronal forme donc un ensemble d'organes élémentaires, appelés *neurones*, dont le modèle mathématique permet d'émuler certaines propriétés du système nerveux biologique, à savoir celles qui permettent un apprentissage adaptatif. Les systèmes eux-mêmes sont adaptatifs, parce que les interconnexions peuvent être *effectrices* ou *inhibitrices*, selon qu'elles renforcent ou atténuent l'information mémorisée et transmise par chacun des nœuds.

Les neurones sont reliés par des connexions dotées d'un indice de poids. Ils réagissent à une stimulation par un phénomène de propagation qui vient réajuster de façon régressive le poids des connexions environnantes. La connaissance nécessaire pour résoudre des problèmes réside dans le poids des connexions. L'apprentissage consiste à ajuster ces poids par l'exposition répétée de différentes stimulations qui affectent les neurones. Schématiquement et en faisant l'économie des équations, le résultat d'une stimulation pour chacun des neurones est calculé à partir de la somme des valeurs résultantes du calcul de la stimulation des neurones associés. Ces valeurs sont à leur tour multipliées par une valeur spécifique, le poids synaptique, associée à chacune des connexions. L'apprentissage s'opère soit de manière dirigée, à partir d'un entraînement sur des exemples choisis, soit de manière non-dirigée, à partir d'une exposition prolongée aux situations concrètes qui se présentent.

Même s'il s'agit d'une reproduction bien imparfaite et trop schématique du raisonnement biologique, les réseaux neuronaux sont utiles. Ils sont d'ailleurs utilisés pour résoudre des problèmes qui exigent que soient reconnues la forme et la classe d'un item dans des situations où la représentation de cet item fait l'objet d'une distorsion. Ce genre de problèmes se rencontre souvent dans la reconnaissance des caractères typographiques et de la parole.

2.5.6. La planification automatique



Le premier programme d'intelligence artificielle reconnu comme tel date de 1956. C'est le *Logical Theorist* de A. Newell, H. Simon et J. C. Shaw. Il s'agit d'un programme qui trouve les preuves de théorèmes mathématiques à partir d'arbres de décisions, en effectuant des sélections successives dans la branche de l'arborescence qui aura le plus de chances d'aboutir à une démonstration valide. L'année suivante, les mêmes concepteurs ont proposé le *General Problem Solver* (GPS) qui généralise la stratégie du *Logical Theorist*. Cette stratégie consiste à essayer d'atteindre le but ou de raffiner une certaine solution à partir d'un processus itératif. Les auteurs voulaient non seulement que le GPS trouve la solution correcte du problème, mais aussi que la trace du raisonnement corresponde étape par étape à celle du raisonnement des sujets humains résolvant les mêmes problèmes.

Cette idée d'un système orienté vers les buts est dérivée d'une théorie économico-organisationnelle proposée par Simon. Suivant cette théorie, les différentes parties d'une organisation ne connaissent et ne traitent que les sous-but, alors que sa partie supérieure dicte les buts à atteindre et coordonne les activités des départements en fonction. Simon établit ainsi une analogie entre l'organisation du travail et l'intelligence en général, c'est-à-dire aussi bien l'intelligence d'une organisation, d'une colonie de fourmis, d'une fourmi et d'un cerveau de fourmi.

Le résultat de ce genre de programmes est constitué d'une série d'actions ou d'étapes à suivre, et de la liste des sous-but. Ces actions doivent réduire la différence entre l'état actuel et l'état qui constitue le but à atteindre, c'est-à-dire la solution du problème posé. Ce programme, appelé *planificateur*, produit des états de connaissance de façon itérative dans un espace de problème donné. Cet espace de problème se voit définir par un état initial, par un but à atteindre qui se décompose lui-même en sous-but, et par un ensemble d'opérations sur les états de connaissance successifs. La mémoire du système est divisée en deux parties. La première est consacrée aux connaissances procédurales, et la seconde à l'espace du problème qui comprend l'état de connaissance courant, l'état du but, les sous-but qui doivent solidairement être atteints pour que le but soit atteint, les sous-sous-but, etc. Les planificateurs sont *progressifs* s'ils cherchent vers l'avant, c'est-à-dire



à partir de l'état initial vers l'état but. Ils sont *régressifs* s'ils cherchent à dériver l'état initial à partir de l'état caractérisant le but, à la manière d'un démonstrateur de théorème.

Durant les années 1970, les planificateurs ont évolué dans plusieurs directions. Entre autres, pour contourner l'écueil de l'imprévisibilité d'un environnement réel, certaines recherches ont développé le concept de *micro-monde*. Un micro-monde est un environnement totalement contrôlé, dont les composants sont décrits en termes d'objets dotés de propriétés. Des robots immatériels y furent développés pour effectuer certaines tâches simples, comme la manipulation de blocs ou d'anneaux sur une tige. Pour effectuer ces tâches apparemment simples, il faut notamment comprendre la géométrie tridimensionnelle et certains aspects de la physique. Proposé par T. Winograd en 1970, le programme SHRDLU (acronyme sans signification) a fait considérablement avancer les planificateurs dans un monde de blocs. Ce système peut être interrogé par un humain, à qui il répondra ce qu'il fait et pourquoi.

2.5.7. Des systèmes experts

À côté des planificateurs dont ils sont proches parents, les systèmes à base de connaissance, nommés aussi *systèmes experts*, veulent capter le raisonnement d'experts pour pouvoir établir des diagnostics ou prendre des décisions dans des situations de la vie réelle. Proposé par T. Shortliffe en 1974, Mycin encapsule une expertise médicale suffisante pour diagnostiquer des infections bactériologiques du sang et suggérer un traitement approprié. Le système expert est un système universel de raisonnement symbolique fondé sur l'inférence. L'*inférence* est l'opération logique d'association qui, à partir de certains faits tenus pour vrais, permet de transférer la valeur de vérité de ces faits sur d'autres. À partir de faits initiaux, le système induit d'autres faits, qui constituent la réponse attendue après de nombreuses itérations.

L'inférence s'opère à partir d'une règle qui, comme les instructions conditionnelles, comporte deux parties. D'une part, il y a l'ensemble de conditions appliqués aux



faits, appelé prémisse. D'autre part, il y a l'ensemble de faits dont la vérité est confirmée si la règle est déclenchée par un filtrage positif de la prémisse en regard des faits déjà avérés. Le raisonnement est *monotone* si aucun retour en arrière n'est possible; il est *non-monotone* s'il est possible de retirer la valeur de vérité à des faits déjà inférés, de renverser des conclusions suite au constat qu'une prémisse était erronée et ensuite de continuer le raisonnement.

Les règles d'inférences sont regroupées en bases de connaissance et constituent, en quelque sorte, l'expertise du système. Remarquons, au passage, le principe de décomposition qui a caractérisé le développement de l'informatique (cf. section 2.5.5). Les instructions conditionnelles du programme deviennent des données structurées en règles, et un algorithme général s'occupe des cycles d'inférences, en déclenchant jusqu'à épuisement les règles pertinentes au vu des faits avérés par les inférences précédentes. C'est pourquoi on parlera de *programmation déclarative*. Cette séparation entre les règles et le moteur d'inférences rend le système facilement programmable. Les règles sont indépendantes les unes des autres. Elles sont locales : on peut en tout temps en ajouter et en changer sans provoquer de réaménagement de l'ensemble, ce qui permet un développement incrémental du système.

2.5.8. La représentation des connaissances

Les systèmes experts focalisent sur la performance de la machine dans des situations *réelles*, où les problèmes se voient généralement mêlés les uns aux autres, et où les informations se révèlent souvent incomplètes ou incertaines. Sur ces deux points, la notion d'*heuristique* développée par G. Polya en 1945 s'avérera rapidement capitale. Les connaissances heuristiques sont tirées non pas d'un savoir scientifique encyclopédique, mais de l'expérience vécue des experts. Ces connaissances acquises, validées et raffinées par la pratique sont colligées par un *ingénieur de la connaissance*, et par la suite découpées, puis structurées et finalement exprimées sous la forme appropriée pour les systèmes experts.



La *représentation de la connaissance* est l'opération qui consiste à formaliser et à organiser la connaissance. Les unités de connaissance encapsulées dans les règles d'inférences peuvent être avantageusement dotées d'une structure itérative interne du type objet — propriété — valeur. En 1975, M. Minsky propose dans un livre intitulé *A Framework for Representing Knowledge* une notion qui permet de représenter les constructions mentales : les *frames*. L'idée est de constituer les primitives de la pensée en regroupant, autour d'un concept, d'une notion ou d'une chose, les connaissances qui y sont typiquement associées, et d'en faire des propriétés de ce concept. Il est possible de spécifier des relations entre les *frames* en faisant d'un *frame* la valeur de la propriété d'un autre *frame*. Ce faisant, le premier *frame* héritera de l'ensemble des propriétés et des valeurs du second *frame*. Ces *frames* jouent le rôle de prototypes; leur structure est utilisée pour compléter les informations ou les faits fournis au système.

2.5.9. Cumul de l'incertitude

Le cadre strict de la logique booléenne s'est avéré trop limité face à la complexité des situations que présente la réalité. Plusieurs dispositifs ont été proposés pour pallier à la loi du tiers-exclu, comme la *logique floue*, qui permet d'utiliser des valeurs symboliques au-delà des valeurs *vrai* ou *faux*, en désignant des qualités plutôt que des quantités telles que *beaucoup*, *peu*, *anormal*, *conforme*, etc. Ces valeurs sont par la suite intégrées à un intervalle continu de valeurs de vérité entre 0 et 1. Les systèmes experts permettent de traiter l'incertitude à un certain degré avec des coefficients. Le moteur d'inférences propage l'incertitude ou au contraire la confiance tout au long de la chaîne inférentielle. Selon la conjoncture, les formules de cumul provoquent soit une atténuation, soit un renforcement du coefficient d'un fait donné. Un renforcement aura lieu lorsqu'un fait est inféré plus d'une fois. Ce principe permet de discriminer entre différentes inférences équivalentes, et ainsi de réduire le nombre et la complexité des règles nécessaires. Il permet aussi de dépasser l'unicité d'un résultat pour obtenir des associations plurielles, mais différenciées et ordonnées. Toutefois, pour les experts, il s'avère dans les faits très difficile d'exprimer sous la forme d'un coefficient numérique le degré de confiance



qu'ils ont dans les relations qu'ils établissent entre des faits vérifiés ou tenus pour vrais et les conclusions qu'ils en tirent. De plus, pour que l'effet discriminant soit optimal, un calibrage des coefficients doit être effectué en fonction du nombre de renforcements qui sont susceptibles de se produire. Or, cela va à l'encontre du principe de modularité des systèmes experts.

2.5.10. Limites des systèmes experts

Les systèmes experts sont fondés sur la logique propositionnelle. Les premiers systèmes experts sont programmés au complet : la base de règles et le moteur d'inférence. Par la suite sont apparus des *générateurs de systèmes experts*, littéralement appelés *coquilles* (ou *shells* en anglais), qui facilitent la déclaration et la gestion des règles. En plus d'un éditeur de la base de connaissances, les générateurs comportent aussi un moteur d'inférences adapté au formalisme de représentation. En marge des générateurs de systèmes experts qui adaptent la logique propositionnelle d'une certaine manière, PROLOG (PROgrammation LOGique) fut proposé par Alain Colmerauer en 1972. Fondé sur la logique des prédicats du premier ordre, ce langage se substitue au générateur de l'un ou l'autre type pour programmer directement des systèmes experts d'un type particulier. En PROLOG, la division du problème initial en sous-problèmes de plus en plus élémentaires s'opère par *unification*. L'unification est une procédure qui consiste à permuter deux arguments lorsque, dans deux clauses, les prédicats sont identiques. Les sous-problèmes sont résolus par unification avec des clauses sans prémisse, dont les arguments sont des constantes. Ce type particulier de clause constitue les faits.

Comme tous les autres programmes informatiques, les systèmes experts ne peuvent manipuler syntaxiquement que les symboles d'où toute signification est totalement exclue. Après une vague d'optimisme débridée, d'échecs et de faillites, les systèmes experts sont largement utilisés, mais uniquement pour une classe de problème requérant des connaissances profondes quoique circonscrites au champ d'expertise. La représentation des connaissances de sens commun, quant à elle, s'est avéré constituer un problème majeur dans la généralisation des systèmes



experts, en raison du nombre, de la diversité et de l'intrication de ce type de connaissances. Un exemple très simple : le feu brûle ou liquéfie alors que le froid gèle. Ainsi, pour réaliser un système expert autonome capable de déterminer comment il faut se vêtir au moment de sortir, il faut qu'il puisse choisir et accomplir l'une ou l'autre stratégie la mieux adaptée au contexte : faut-il regarder par la fenêtre ou regarder la chaîne météo à la télévision? Il faut ensuite détecter s'il y a une fenêtre et se diriger vers elle, il faut allumer le poste de télévision et syntoniser la bonne chaîne, etc.

2.5.11. Des agents intelligents

À la fin des années 1980, les systèmes experts et la programmation orientée objet se croisent pour donner une société d'agents. En 1987, Marvin Minsky propose *The Society of Mind*, ouvrage consacré à la description théorique du cerveau conçu comme une collection, ou plutôt comme une colonie d'agents coopérants. Les *agents intelligents* constituent la classe la plus avancée d'automates digitaux. Un agent est une entité informatique qui déploie une certaine activité en fonction de certains buts. On les dit *intelligents* parce qu'ils ont la capacité d'apprendre, qu'ils sont dotés d'une capacité sociale et d'un haut degré d'autonomie. Ceci dit, à ce jour, aucun agent ne possède l'ensemble de ces caractéristiques. Contrairement aux systèmes experts que l'on voyait déjà remplacer les experts humains, les agents intelligents ont pour mission d'assister les humains, ceux-ci étant considérés comme des agents également inclus dans le système.

L'agent est autonome. Son activité a lieu sans supervision ni intervention humaine; il exerce une forme de contrôle sur ses actions; il peut examiner ses objectifs, nommer ses intentions, planifier ses actions et modifier ses objectifs au fur et à mesure du déroulement des actions. L'agent est en interaction constante avec son environnement informatique et a la capacité d'intervenir dans les processus en cours. Parmi ces processus, il y a le système d'exploitation de l'ordinateur, des bases de donnée et toutes sortes d'applications. L'agent interagit également avec d'autres agents et, éventuellement, des êtres humains, grâce à un langage de communication inter-agents. Il est capable de coordonner ses actions par rapport à



un utilisateur ou un autre agent. Par exemple, un agent de sauvegarde qui rencontre un autre agent peut s'entendre avec lui de manière à partager le travail efficacement par rapport aux contraintes qui ont été spécifiées.

L'agent perçoit son environnement informatique et réagit en répondant dans les temps impartis aux changements et aux événements qui y surviennent. L'agent a la capacité d'apprendre, peut acquérir des connaissances, des informations ou des habitudes et les utiliser lorsqu'il réagit et interagit. Schématiquement, l'agent est doté des composants suivants: une base de connaissance, un module d'apprentissage, un moteur d'inférences, un moteur d'interactions et une table de réactions.

Les agents sont fabriqués avec des outils de création d'agents, qui fournissent un protocole et des modalités d'édition. La plupart des agents intelligents fonctionnent sur le réseau Internet. Les agents sont dits mobiles lorsqu'ils peuvent se déplacer d'une machine à une autre. Pour migrer, un agent mobile transfère son cœur et ses données sur le nouveau site, où il continue à s'exécuter. À l'issue de la migration, le processus de la machine initiale est détruit par une commande du nouveau processus.

La plupart des agents intelligents sont affectés à la recherche d'information dans l'Internet. Plusieurs agents peuvent coopérer pour répondre à une requête d'information, chacun étant doté de la connaissance nécessaire pour fouiller dans les bases de données et les autres formats de fichier, et pour filtrer le bruit causé par des ambiguïtés linguistiques ou sémantiques. Ces agents peuvent aussi, une fois les gisements de données identifiés et cartographiés, faire régulièrement état des changements éventuels et ne rapporter que les nouvelles informations pertinentes pour une certaine requête d'information.

La conception de systèmes informatiques à l'aide d'agents intelligents permet d'associer plusieurs classes ou plusieurs groupes spécialisés d'agents aux différentes fonctionnalités du système. Ce faisant, on améliore la modularité, la flexibilité et l'extensibilité du système. Les agents peuvent se voir attribuer chacun un rôle différent dans un système, et ainsi se partager les tâches, qu'ils exécutent en



parallèle, tout en partageant l'information en cours d'exécution. Réguler la coordination des agents devient alors capital.

2.6. Les automates médiatiques

Pour clore cet examen systématique des différents automates digitaux immatériels, examinons quelques automates qui interviennent dans plusieurs manifestations des médias interactifs.

2.6.1. La navigation Web

L'hypertexte est une technologie de l'information qui permet d'opérer une transition associative entre un point d'un certain document vers un autre point du même document, ou de tout autre document se trouvant à l'intérieur d'un espace informationnel donné (cf. section 1.6.5). Pour passer d'un document à un autre, on active un lien mis en évidence typographiquement ou iconiquement; *naviguer* est le terme technique qui désigne cette action. L'hypermédia, dérivé de l'hypertexte, désigne une forme de structuration basée sur des associations d'éléments audiovisuels et informationnels et dont l'accès est laissé à l'utilisateur. Les pages Web sont reliées entre elles et à d'autres services par un mécanisme de type hypertextuel.

Le logiciel client est le logiciel à l'aide duquel l'interacteur formule ses requêtes au serveur pour obtenir des documents ou des services, comme le courriel ou le téléchargement de fichiers. Ce logiciel client comporte un premier automate qui transforme en écrans graphiques les fichiers en format texte, comportant des commandes en HTML entre balises. Schématiquement, cet automate traite le fichier caractère par caractère, du premier au dernier. Si le caractère est une balise ouvrante « < », l'automate ouvre un accumulateur pour les caractères suivants jusqu'à ce qu'une balise fermante « > » soit rencontrée. Puis l'action désignée par la balise est exécutée: alinéa, insertion d'un fichier d'image, etc. Lorsque la commande s'applique à un ensemble de caractères, l'automate doit détecter la balise qui marque le début et la fin de la séquence, indiquée par le « \ » qui précède



l'identificateur de balise; l'effet demandé et détecté à partir de la balise de début de séquence — affichage en gras, changement de police, justification — ne sera généré que lorsque l'automate a trouvé la balise de clôture.

L'automate qui assure la navigation intervient, dans un premier temps, lorsque les balises appropriées sont détectées (par exemple : Bibliographie). Dans un second temps, la séquence de caractères entourée par les balises de lien est mise en évidence; habituellement, elle est soulignée ou mise dans une couleur particulière. Lorsque l'utilisateur clique sur le texte ainsi mis en évidence, il active l'automate qui va alors effectuer plusieurs opérations. Tout d'abord, l'automate décompose l'adresse fournie entre guillemets en trois segments : type de service, serveur et document. S'il s'agit d'un service Web, dénoté par « http », l'automate demande une page web dont le fichier est nommé « index.html » au serveur, ici « uqam.ca ». S'il s'agit d'un autre service, l'automate passe la main à l'automate adéquat et démarre un chronomètre. Si l'automate ne reçoit pas de réponse de la part du serveur visé dans le délai imposé, il affiche un message d'échec de la requête. Quand un contact a été établi, l'automate assure la réception et la mise en ordre des lots de paquets (section 1.7.3) de chacun des constituants de la page demandée.

Somme toute, la navigation est une tâche simple assez simple. Toutefois, pour accomplir cette tâche, il faut tout de même recourir à un automate suffisamment détaillé pour pouvoir prendre en compte toutes les éventualités.

2.6.2. Les moteurs de recherche

Le moteur de recherche est une technologie de l'information qui permet de formuler une requête de documents, non pas, comme dans le cas de la navigation, à l'aide de l'adresse précise de ces documents, mais à partir de mots qu'ils contiennent (cf. 1.6.3). Cette tâche requiert un double mécanisme: l'indexation des documents et la recherche en tant que telle.

Il est impensable d'effectuer une recherche en parcourant du début à la fin chacun des documents de l'espace d'information. Le délai nécessaire à la recherche varierait



trop en fonction du nombre de documents. Une recherche efficace exige que les documents aient été préalablement indexés. Un index forme une espèce de tableau à deux colonnes, la première contenant le mot, la deuxième représentant la liste des références des pages où le mot apparaît. L'automate d'indexation prend en entrée une page Web et, éventuellement, les pages qui sont reliées à celle-ci, et ainsi de suite. Le document est traité mot par mot du début à la fin. Un mot est une suite de caractères qui se trouve entre deux séparateurs, espaces ou signes de ponctuation. D'abord, les balises sont séparées des mots en tant que tels. Si le mot n'est pas dans l'index, une nouvelle entrée est créée; s'il existe déjà, la référence au document est ajoutée à la suite des autres.

Une requête simple est composée d'un mot; une requête complexe est composée de plusieurs mots. Dans le second cas, le type de relation souhaité entre chacun des mots doit être spécifié, soit la co-présence des mots, soit la présence d'un mot en l'absence d'un autre. Une requête complexe peut aussi imposer un certain intervalle entre les différents mots devant être présents, soit la contiguïté immédiate si c'est une expression figée, soit la présence à l'intérieur d'un intervalle de quelques mots (pour dépister, par exemple, les occurrences où un adjectif s'intercalerait entre les deux mots formant l'expression recherchée, etc.).

Le moteur de recherche est un automate traitant à partir d'un serveur les requêtes qui doivent d'abord être saisies sur un formulaire, et ensuite acheminées. La première opération exécutée par cet automate consiste à accéder par l'index aux listes d'adresses pour chacun des mots de la requête. Ensuite, si la requête est multiple, l'automate traite les opérations logiques : intersection, union, négation sur les listes d'adresses correspondant à chacun des mots. Puis la liste d'adresses est ordonnée en fonction de critères arbitraires mais reconnus pour leur efficacité : la fréquence d'apparition du mot dans un même document, la date d'inscription du document dans l'index, la présence du mot dans un endroit stratégique du document (comme le titre). Enfin, une page de résultats est composée dynamiquement ; celle-ci comporte pour chacun des fichiers jugés pertinents, l'adresse et certains mots ce qui facilite l'épuration des résultats par l'utilisateur.



Les moteurs d'indexation des documents et les moteurs de recherche qui leur sont associés sont des automates qui effectuent des tâches assez faciles et hautement répétitives. Toutefois, l'ampleur de l'espace documentaire, constitué de millions de pages, et le nombre d'utilisateurs qui peuvent simultanément soumettre des requêtes sont autant de facteurs qui peuvent rendre ces tâches, a priori simples assez complexes, suffisamment en tout cas pour que le système ne soit pas toujours sans faille.

2.6.3. Le tableau, mécanisme du puzzle

Le dernier exemple d'automate digital abordé est celui qui traite du fameux puzzle de neuf cases, dont une vide, et qui présente un dessin dans le désordre. La tâche consiste bien évidemment à reconstituer l'image dans le bon ordre en déplaçant une case à la fois, obligatoirement choisie parmi les cases adjacentes à la case vide. En voici un exemple à partir d'un masque africain :



L'automate effectue les opérations suivantes. Au début du jeu, les neuf fragments d'image sont distribués au hasard sur la planche de jeu. La case vide est aussi tirée au hasard et le fragment d'image dans la case correspondante est retiré. C'est à l'utilisateur de jouer. Il clique sur l'une des cases ; l'automate doit vérifier si une case adjacente est vide. Les cases adjacentes sont répertoriées sous forme d'un tableau associatif où, pour chacune des cases, les adjacences sont listées. Ainsi, la case 1 en haut à gauche a pour cases adjacentes la case 2 à sa droite et la case 4 en bas ; la case 5 au centre du tableau a les cases 2, 4, 6, 8 pour voisines. Une fois dressée la liste des cases adjacentes, celles-ci sont testées une à une de manière à trouver la case vide ; lorsque celle-ci est repérée, l'automate substitue le carré noir



et le fragment d'image, donnant ainsi l'impression que le fragment a avancé. Si, en revanche, aucune case adjacente n'est vide, la substitution ne peut avoir lieu.

2.7. Bibliographie

Beaune, Jean-Claude, *L'automate et ses mobiles*, Paris : Flammarion, 1980, 469 p.

Breton, Philippe, *Une histoire de l'informatique*, Paris : Éditions du Seuil , 1990, 261 p.

Breton, Philippe, *À l'image de l'homme : du Golem aux créatures virtuelles*, Paris : Éditions du Seuil, 1995, 187 p.

Chapuis, Alfred et Edouard Gelis, *Le monde des automates : étude historique et technique*, Genève : Slatkine, 1984 (2vol.) Réimpression de l'édition de Paris, 1928.

Krzywkowski, Isabelle, *L'homme artificiel : Hoffmann, Shelley, Villiers de l'Isle-Adam*, Paris : Ellipses, 1999, 192 p.

Maingot, Eliane, *Les automates*, Paris : Hachette , 1959, 95 p.

Menzel, Peter ; D'Aluisio, Faith, *Robo sapiens : une espèce en voie d'apparition*, Paris : Autrement, 2001, 245 p.

Robinet, André, *Le défi cybernétique : l'automate et la pensée*, Paris : Gallimard , 1973, 232 p.

Wurster, Christian, *L'histoire illustrée des ordinateurs*, Köln : Taschen, 2002. 327 p.

