

## Informatique et simulation

Francis ROUSSEAU  
Professeur d'informatique à l'Université de Reims  
Délégué CNRS à l'IRCAM

tél. 01 46 33 60 03  
20, rue de Condé, 75006 Paris  
[francis.rousseau@ircam.fr](mailto:francis.rousseau@ircam.fr)

### Résumé

Un ordinateur est une *machine de Turing* particulière, et la conception d'un programme informatique interactif vise la réalisation *d'agents artificiels coopératifs* susceptibles d'être investis d'esprit par des utilisateurs humains.

Ici, le terme “agent artificiel” ne renvoie pas aux agents de *la programmation par objets* connus des programmeurs, mais bien davantage à une métaphore de sens commun. Selon l'intuition radicale de Turing en effet, l'investissement d'esprit se déploie par le *dialogue*, qui seul ouvre à la personnification d'une altérité : le propre d'un sujet, c'est de reconnaître la subjectivité de l'autre.

L'investissement de *confiance* d'un dispositif artificiel peut certes s'établir sur la base d'une *interaction* fruste, comme celle qui caractérisait les premiers programmes de calcul, lorsqu'un prompt apparaissant sur l'écran constituait à la fois une invitation à faire sens du résultat précédemment affiché et une nouvelle disposition de la machine à communiquer. La métaphore dialogique est ici celle de l'oracle omniscient, omniprésent et, pourvu que les formes rituelles de la consultation soient scrupuleusement respectées, idéalement rationnel. La puissance évocatrice de cette métaphore ancestrale est telle qu'elle noue implicitement l'attention, l'intention et la confiance de l'utilisateur venu consulter.

Cependant, *passer pour avoir de l'esprit* n'est pas ordinairement attribué aux artefacts. Les informaticiens le savent bien, qui dotent leurs machines en réseau de programmes interactifs inspirés de métaphores de la présence et de la connaissance toujours plus ingénieuses, dans le but d'ouvrir ce mystérieuse pouvoir à leur réalisation.

Aussi la programmation entretient-elle un rapport essentiel et natif avec *la simulation*, indépendamment du fait que l'exécution du programme réalise ou non *une simulation de tel ou tel phénomène* (dans cette phrase même, “simulation” évoque concurremment la tragédie grecque et l'art de la guerre d'une part, les automates de Vaucanson et les modèles numériques de l'autre).

Il convient cependant de mettre en question cette prétendue indépendance.

## 1° Simulation et paradoxe du comédien

Considérons le “paradoxe du comédien” écrit par Diderot en 1773. Sous forme de dialogue, Diderot pose la question : la nature, sans l'art, peut-elle former un grand comédien ? Le premier interlocuteur (Diderot lui-même) arrive vite à l'essentiel du dialogue; pour lui les qualités essentielles d'un bon comédien sont le *jugement* et la *pénétration*; il n'a nul besoin de *sensibilité*; bien mieux, la sensibilité peut lui être contraire.

Est-il nécessaire de le prouver ? Le jeu d'un acteur sensible, “ qui joue d'âme ”, serait inégal; celui qui “ joue d'étude, de réflexions ” se forme un modèle idéal du rôle, d'après lequel tout sera mesuré, combiné, appris dans sa tête. Mlle Clairon réalise le type de la comédienne dont le jeu est réglé par ce modèle idéal; Mlle Dumesnil, au contraire, ne fait à l'art qu'une place très petite; elle s'abandonne à la nature, à *l'enthousiasme*. Théorie qui pourrait s'appliquer aux grands poètes, qui sont les êtres les moins sensibles.

Voici une série de constatations susceptibles d'étayer cette thèse, selon Diderot.

- il s'opère, dans le grand acteur, un dédoublement : il agit sur la scène sans rien sentir; il n'est pas le personnage qu'il joue, mais il le joue si bien que vous le prenez pour tel; l'illusion n'est que pour les spectateurs. Le vrai, sur la scène, c'est la conformité des actions, des discours, de la figure, de la voix, du mouvement, du geste avec un modèle idéal imaginé par le poète et souvent exagéré par le comédien. Les conventions sont utiles et même indispensables au théâtre : on ne meurt pas sur la scène comme en meurt en réalité. La nature a des moments sublimes; encore faut-il les saisir de sang-froid;
- c'est dans la vieillesse, quand ils ont acquis une longue expérience, que les acteurs sont les meilleurs; la tête est plus calme, l'âme se possède;
- mille petits faits montrent que, loin d'être possédé par son rôle, l'acteur le possède. Garrick était capable de passer sans transition d'une scène de comédie à une scène de tragédie. Il en est de même dans la vie. L'homme sensible obéit aux impulsions de la nature, et ne rend que le cri de son cœur; quand arrive que, lorsque le comédien s'éveille en nous, nous obtenions ce que la sincérité ne nous avait pas permis d'obtenir;
- un moyen de jouer petitement, mesquinement, c'est d'avoir à jouer son propre caractère. Le propre du comédien consiste, au contraire, à sortir de son propre caractère et à copier tous les autres. Mlle Clairon avait imaginé un grand fantôme et s'y conformait avec génie, si bien qu'en l'entendant dans une de ses pièces, Voltaire s'écriait : “ Est-ce bien moi qui ai fait cela ? ” Après maintes digressions, Diderot, par la bouche du premier interlocuteur, définit ainsi le comédien : “ Un grand comédien est un pantin merveilleux, dont le poète tient la ficelle et auquel il indique la véritable forme qu'il doit prendre ”;
- dans la vie courante, les comédiens auxquels on prête tant de sensibilité, en montrent peu. On les voit souvent rire en dehors de la scène, on les voit rarement pleurer. En somme, les caractères ne sont si bien rendus par les comédiens que parce qu'eux-

mêmes n'en ont aucun. Grands sur la scène, ils sont petits et bas dans la société : et si la profession de comédien n'est ni honorable ni honorée (sic), le fait est imputable aussi à l'état général des mœurs.

## **2° Simulation et informatique : le champ opératoire**

L'informatique est un *champ disciplinaire* qui n'existait pas avant la seconde Guerre mondiale et qui fait néanmoins carrière polymorphe et fulgurante aujourd'hui, conférant le rang de *phénomène de société* notoire à son développement tentaculaire. “Nouveauté disciplinaire” et “percolation spectaculaire des usages” tendraient ainsi à caractériser l'informatique comme ayant nativement partie liée avec son déploiement spécifique.

Aussi la question “ Qu'est-ce que l'informatique ? ” a-t-elle pulvérisé en une myriade de réponses pratiques et de préconisations tactiques, siégeant au lieu même des expérimentations singulières. Manifestement, l'informatique invite davantage à des mises en œuvres innovantes et créatives liées à des domaines d'activités humaines particulières qu'à des considérations abstraites sur son essence. Et il semble bien qu'on “n’y comprenne quelque chose” en informatique qu'à la seule condition d'en maîtriser une région technologique ou d'en coproduire un secteur de marché.

Cela tient-il à *l'immaturité passagère* de la discipline scientifique, encore incapable de synthétiser son individualité ?

Il est vrai que cette discipline scientifique, mal détournée de sa gangue originale, reste encore fortement soumise aux contingences de son avènement : et les lectures épistémologiques de l'informatique demeurent rares, fragmentaires et cantonnées, quand la thématization scientifique se démêle malaisément des luttes institutionnelles qui en tracent âprement les figures, et que la cartographie technologique se confronte à l'explosion des marchés pour se confondre avec leur diversification ...

Cela tient-il à la forte *variété des applications* possibles des technologies de l'information et de la communication, que leur mobilisation accélérée n'est pas encore parvenue à saturer ?

Il est vrai que les prégnantes Nouvelles technologies de l'information et de la communication (NTIC) transforment nos pratiques du commerce, du service, de l'école et de la démocratie, le phénomène de société tendant à s'autonomiser de son sol scientifique pour devenir l'enjeu disputé de politiques industrielles, commerciales et culturelles, voire éducatives ou familiales, sectorisant toujours davantage les applications informatiques en les dédiant à des secteurs d'activité ...

Et si cela tenait à l'informatique de manière bien plus profonde qu'il n'y paraît, pour en constituer un trait essentiel et radical, que le sens commun aurait spontanément intuitionné ? L'informatique ne serait pas seulement un ensemble de pratiques humaines, mais également une *interprétation* des pratiques humaines.

La question mérite d'être posée dès les prémices d'un travail philosophique en informatique car, par delà les options méthodologiques qu'elle implique, elle conditionne le devenir même de l'investigation en prescrivant son mode d'engagement.

En effet, oublier que l'informatique interprète les pratiques humaines reviendrait à tenter d'établir la vision synthétique d'une discipline en faisant effort pour en rassembler les manifestations éclatées sous l'égide d'un département scientifique, sans avoir maîtrisé le risque de ne parvenir qu'à ajouter un peu à la confusion épistémologique.

Une telle entreprise, en imputant à l'avance tout échec à *l'immaturité* disciplinaire, participerait en définitive à la consolidation de la posture théorique dominante : l'échec résiderait définitivement sous la responsabilité de l'impatience, *faute d'avoir su différer* la synthèse scientifique jusqu'à une hypothétique maturité que finirait toujours par conférer "le recul" (cf. le "Ferdynurke" de Gombrowicz).

Certains contournent adroitement la difficulté en préconisant de ne jamais prendre le risque de normer les possibilités créatrices de ce nouvel univers qu'est l'informatique, la priorité étant selon eux d'éveiller la vocation des pionniers et de motiver les créateurs de valeur et d'emploi.

Mais convenons que la raison philosophique rejoint ici la stratégie, et que s'efforcer de synthétiser *hic et nunc* le champ disciplinaire *avec* le phénomène de société NTIC sous une spécification scientifique relèverait d'un projet utopique et finalement dogmatique, qui n'éclairerait pas nécessairement ceux, et ils sont de plus en plus nombreux, qui souhaitent mieux *comprendre* l'informatique.

Une fois admis que la synthèse thématique n'est pas la méthode appropriée, reste à décider quelle approche mobiliser pour contribuer à une meilleure intelligibilité de l'informatique, de ses enjeux et de ses possibilités, en conférant un statut philosophique à l'investigation.

### **3° Une approche calquée sur celle d'Hans Jonas en biologie philosophique**

Lorsque Hans Jonas cherche à élaborer une biologie philosophique, il est confronté à un problème de méthode assez voisin de celui que nous venons d'évoquer, alors même que la biologie scientifique et l'existentialisme phénoménologique se combattent sur la question de la vie sans se livrer jamais bataille concrète, faute de théâtre d'opérations possible (cf. l'ouvrage "Le phénomène de la vie" pour un argumentaire plus précis).

Et c'est en organisant tel ou tel combat singulier, dont le périmètre a priori n'est pourtant pas déterminé ni les règles établies parfaitement, que le dépassement se dessine, autour de concepts-clés intuitionnés par l'investigateur comme permettant précisément cela (c'est ainsi que pour Hans Jonas, le métabolisme est un de ces concepts-clés).

Quant à nous, il nous faut mobiliser notre expérience d'informaticien pour trouver le ou les *concepts-clés* qui ouvriront à une compréhension philosophique de l'informatique, sur le mode de la fertilisation croisée et *réciproque*. En d'autres termes, il nous faut exhiber le

vis-à-vis conceptuel de l'informatique qui s'impose, sans présumer de son lieu de résidence actuel, qu'il soit dans l'espace circonscrit par les descriptions habituelles de l'informatique ou hors de celui-ci.

Prenons acte des candidatures spontanées : La notion de programme ? La notion d'algorithme ? La notion de modèle exécutable ? La théorie de la calculabilité ? La notion d'interaction personne-machine ? L'intelligence artificielle ? La représentation des connaissances et l'hypothèse du "niveau des connaissances" d'Alan Newell ? Le fameux test imaginé par Alan Turing ? Un programme de recherche devrait sans doute exposer toutes ces notions, mais il est de toutes façons nécessaire de pressentir les concepts-clés, et si possible de façon engagée, car ils constitueront les biais autour desquels l'investigation s'enracinerait et se déploierait.

L'enquête peut tout naturellement commencer par interroger la notion de *programme informatique*.

Les programmes informatiques sont des textes interprétés par les programmeurs qui les écrivent mais ils sont aussi, de façon homonyme bien qu'en un sens très différent, interprétés par d'autres programmes qui les réalisent ultimement comme calcul effectif.

C'est ainsi que des modèles formels, à condition de posséder certaines propriétés, peuvent être opérationnalisés par le truchement d'un ordinateur, cette machine qui concrétise la machine abstraite de Turing dans une architecture physique du type de celle proposée par Von Neumann.

Écrire des programmes informatiques est une activité soumise à cette double contrainte de rédaction, faisant droit d'une part à la relecture critique par un programmeur, et exigeant d'autre part le respect de règles syntaxiques et algorithmiques garantissant la légalité du programme en phase de compilation et sa terminaison en phase d'exécution.

Reste que cette double contrainte est surdéterminée par la *visée modélisatrice du programmeur*, qui prétend produire un artefact dont l'opérationnalisation modélise un phénomène plus ou moins complexe.

Ainsi, si programmer consiste à la fois à produire un artefact et à modéliser un phénomène, l'art de la programmation consiste à penser ces deux activités comme une seule, consistant à réaliser des modèles effectifs de phénomènes donnés, de façon prédictive et traçable.

Et les "hackers", ces mordus de la réalisation de programmes, sont sans doute fascinés par la possibilité de faire converger tant de contraintes hétérogènes au travers d'une activité vécue comme cohérente, spécifique et irréductible.

Mais revenons sur la programmation comme cadre scientifique théorique. Certains remarqueront à juste titre que beaucoup de modèles mathématiques ne sont pas "computationnels", au sens où ils sont théoriquement non explicites par des algorithmes qui terminent.

D'autres ajouteront que le caractère formel d'un ensemble de descriptions ne manifeste en rien l'aptitude de cet ensemble à faire fonction de modèle utile.

D'autres encore s'étonneront que des programmes interactifs très simples semblent modéliser adéquatement des activités complexes avec une facilité déconcertante et un succès incontesté, comme ceux qui implémentent la *métaphore* du bureau, des dossiers et des fichiers à usage des employés de bureau.

Ils ont également raison, et Turing lui-même le présentait en proposant son fameux test qui devait marquer l'histoire de l'informatique : l'interactivité personne-ordinateur est une sorte de dialogue allégorique, d'un genre nouveau, dont la maïeutique attend toujours son Socrate.

Mais si les ordinateurs ne peuvent opérationnaliser que des formalismes computationnels et nous condamnent ainsi à ne rien dire des autres, si les modèles utiles ne sont pas nécessairement formels et si les descriptions formelles ne sont pas nécessaires pour élaborer un modèle interactif utile, comment fixer un cadre théorique robuste pour décrire la programmation ?

Cela exige notamment de comprendre comment l'interactivité du dialogue personne-ordinateur facilite l'émergence de modèles utiles à partir d'effectivités computationnelles parfois très simples, l'ordinateur apparaissant comme un interlocuteur artificiel dont la bonne volonté systématique est telle qu'elle incite l'utilisateur à la persévérance, et le conduit subrepticement à opérer une personnification de la machine.

Si les programmes informatiques (nécessairement formels) peuvent parfois donner lieu à des simulations qui participent d'une certaine façon (pour l'heure mystérieuse) à l'effectivité de modèles utiles, c'est parce qu'un modèle utile est *utile à quelqu'un*, et que le phénomène à modéliser n'est pas extérieur à l'usager du modèle.

Des symboles sensés dénoter des connaissances élémentaires sont manipulés sans présomption de leur sens par des procédures se réduisant ultimement à du calcul, mais leur interprétation contextuelle, réputée possible, reste à la charge de la personne : tout le problème consiste alors à faire en sorte que le "réputé possible" ne se démente pas, la machine étant ainsi supposée proposer du sens nouveau par variation contextuelle raisonnable (logique) du sens ancien.

C'est ainsi par exemple que la répétition du *même*, si facile à produire à l'aide d'algorithmes, peut se manifester à l'usage comme *coopération interactive*, si tant est que la personne reconsidère à nouveaux frais la dernière production de la machine pour en paramétrer la prochaine. C'est de cette façon que fonctionnent actuellement la plupart des outils de recherche documentaire, le système artificiel donnant plus à penser qu'il ne prétend penser.

Les programmes informatiques sont souvent caractérisés par les programmeurs comme par les usagers au travers de la simulation que leur exécution est censée réaliser. On en conviendra, la notion de *simulation* est candidate à permettre la mise en perspective à la

fois de la discipline *et* du phénomène informatiques : elle peut constituer légitimement, à ce titre, le concept-clé que nous cherchions.

Non pas que le lieu de la simulation soit suffisamment lumineux pour éclairer de lui-même le clair-obscur informatique : la référence à la simulation, bien que traversant toute la philosophie et constituant un objet vénérable à ce titre, n'en est pas plus claire pour autant quand elle en vient à rencontrer singulièrement l'informatique.

Au contraire, en rapprochant les deux notions également problématiques d'informatique et de simulation, nous comptons bien tirer profit philosophique au lieu exact de la rencontre, avant d'en tirer conséquence alentour. En d'autres termes, en les confrontant là même où elles prétendent se donner rendez-vous spontanément, nous visons le dépassement des cadres habituels d'appréhension de l'informatique et, respectivement, de la simulation.

La méthode proposée signe l'approche envisagée : il s'agira d'un *processus constructif de modélisation*, alimenté par deux points de vue contradictoires. Et il faudra redonner à la simulation toute sa charge de sens commun (simuler, c'est feindre, faire semblant, faire comme si, faire paraître comme réelle une chose qui ne l'est pas, offrir l'apparence, imiter, déguiser un acte sous l'apparence d'un autre), pour comprendre chemin faisant en quoi la simulation informatique différerait de la simulation au sens que Diderot donne à ce terme dans son texte "paradoxe du comédien" ou que Fink lui donne dans "Le jeu comme symbole du monde".

Notre enquête sur la simulation nous conduira ainsi de la tragédie grecque à la question du *continu* et du *discret*, en passant par l'art de la guerre et les automates de Vaucanson.

#### **4° L'enquête préliminaire**

On vise l'éclairage réciproque de l'informatique en tant qu'elle constituerait une *possibilité inédite* de réaliser des simulations, et de la simulation en tant qu'elle se manifesterait singulièrement en *inspirant le développement* de l'informatique. Au fond, il s'agit de décider si les simulations informatiques inaugurent un genre nouveau de simulation.

Pour parvenir à cette fin, il faudrait mettre au point un mode de commentaire de l'informatique qui ne soit ni strictement historique, ni strictement scientifique, ni strictement technologique, se rapprochant sans doute de *l'herméneutique matérielle* proposée par Bruno Bachimont dans sa thèse de doctorat d'épistémologie.

Ici, on aborde successivement quelques lieux constitutifs de l'informatique, autant pour mettre au point la méthode que pour familiariser les auditeurs des Journées de Rochebrune avec quelques uns des fondements révélateurs de l'épopée informatique : l'intelligence artificielle (IA), les modèles multiagents, la notion d'ontologie en acquisition des connaissances, ainsi que l'informatique multimédia.

#### L'intelligence artificielle et ses métaphores

L'intelligence artificielle a constitué d'emblée l'enjeu du programme informatique originel parmi les Alliés vainqueurs de la seconde Guerre mondiale, qui souhaitaient se donner les

moyens stratégiques de gagner la guerre froide en s'appuyant sur des recherches multidisciplinaires en philosophie analytique, en linguistique, en théorie de l'information, en psychologie cognitive et en mathématique.

L'organisation du programme ressemblait à celle qui avait permis de triompher à Los Alamos en parvenant à manufacturer la bombe atomique, concentrant les chercheurs, les moyens financiers et les effets d'annonce stimulants. Au fond, "intelligence" renvoie au *renseignement* comme le signifie le vocable anglo-saxon.

Il s'agissait de prolonger les brillantes prestations de décodage, par les équipes de Turing, des messages échangés par les bâtiments sous-marins de l'Axe après cryptage par ENIGMA. Pour cela, il fallait maîtriser l'explosion combinatoire caractéristique de la Théorie des jeux, traiter automatiquement le Langage naturel écrit et parlé, surmonter les difficultés de la Reconnaissance des formes par des moyens ingénieux et adaptés aux problèmes spécifiques de l'acoustique, de la vision ou du toucher.

Des langages de programmation allaient voir le jour (LISP inventé par McCarthy en 1956), ainsi que des systèmes de résolutions de problèmes à partir de descriptions formelles (le "General Problem Solver" de Simon et Newell en 1958), des démonstrateurs automatiques de théorèmes qui allaient devenir les fameux Systèmes experts, sans oublier des Systèmes apprentis utilisant parfois des Logiques étendues ou spécifiques.

Au fond, l'intelligence artificielle institue la discipline qui vise la conception de programmes manipulant des symboles interprétables en langue. Ces symboles sont investis de sens dès leur introduction dans leur contexte originel. Les programmes manipulent formellement ces symboles, à l'aide de modèles logiques, et leur réinterprétation en langue dans le nouveau contexte conduit heuristiquement à une suggestion plausible de sens.

Lorsque les manipulations sont interactives et que les dialogues de contrôle font mine de porter sur le sens interprétatif des symboles, l'investissement d'esprit du programme par l'utilisateur peut surprendre (cf. l'expérience de Weizenbaum avec le programme ELISA).

Les résultats furent intéressants au plan technologique et stimulants au plan scientifique, même si les promesses initiales n'ont jamais été tenues. Plus étonnant, la démarche a rencontré d'importantes préoccupations de la philosophie analytique et a suscité des propositions et des controverses en Théorie de l'esprit, dépassant parfois le périmètre de la science cognitive naissante.

Cependant, l'intelligence artificielle a toujours privilégié l'expression langagière, au détriment de l'impression sensorielle.

C'est ainsi que le couple informatique / philosophie analytique a fini par rencontrer la philosophie continentale, aux Etats-Unis même, sous le visage de la philosophie de Heidegger et de l'herméneutique de Gadamer.

On étudiera en particulier les critiques adressées par Hubert Dreyfus et Terry Winograd à l'Intelligence artificielle dans les années soixante-dix : le choc fut très violent et les chercheurs scientifiques occidentaux, heurtés de plein fouet par des controverses sans



avoir toujours la culture philosophique suffisante pour les circonscrire, ont souvent fait effort pour redéfinir leur champ disciplinaire.

Nous essayerons de comprendre les rapports qu'entretiennent l'IA et la simulation.

### Principe de rationalité, objets, agents et acteurs

Au plan de la méthode, la programmation a d'abord été marquée par la scolastique : l'écriture des programmes était subordonnée à leurs conceptions globale puis détaillée (on peut faire le parallèle avec le plan du texte traditionnel, sans lequel il fut longtemps inconcevable d'écrire la moindre ligne), elles-mêmes subordonnées à l'analyse et la spécification des problèmes à modéliser (cette étape étant à rapprocher du positionnement stratégique des textes littéraires, visant à préciser l'effet qu'on en attend sur les lecteurs).

Par la suite, comme pour le développement des outils de "traitement de textes", des environnements techniques de programmations sont apparus qui ont permis au programmeur de s'affranchir de la tyrannie du "cycle en V" (analyse et spécification du besoin -> conception globale et détaillée -> réalisation -> tests unitaires et tests d'intégration -> validation de la satisfaction des besoins) et des paradoxes engendrés par sa stricte observance. La programmation devient vite une activité concurrente, davantage incrémentale et créative.

De la même manière que l'usage des logiciels de "traitements de textes" a transformé l'acte d'écriture en abolissant incidemment la rupture méthodologique fond/forme, ou encore que l'usage des outils de présentation d'information multimédia a généralisé la notion d'arrière-plan graphique issue du théâtre et de l'opéra, la programmation par objets a ouvert une transformation en profondeur de l'activité de programmation elle-même, en facilitant l'usage de prototypes et de maquettes pour préciser progressivement la spécification des besoins et l'adéquation des solutions.

Là encore, nous tenterons de comprendre les rapports entretenus par la simulation et la programmation par objets. Nous nous appuyerons sur des situations pédagogiques de *mise en scène* de textes de l'ancien testament, pour simuler les prises de décision des différents acteurs engagés dans les scènes bibliques (par exemple, nous dirons quelques mots sur "Suzanne au bain", extrait du Livre de Daniel).

### La notion d'ontologie en acquisition des connaissances

La nécessité de concevoir des *Systèmes experts* a rapidement conduit les informaticiens à s'interroger sur la notion de connaissance.

En effet, les Systèmes experts furent conçus comme des Systèmes à bases de connaissances exploitées par des moteurs d'inférences, sur la base de résultats de Logique mobilisant le Principe de déduction (Modus ponens) et généralisant aux connaissances la traditionnelle séparation des données du traitement, préconisée en programmation.

Pour extraire la connaissance des experts, les ingénieurs cogniticiens œuvraient en amont de la conception des systèmes à l'aide de techniques d'interview ou de questionnaires,

jusqu'à ce qu'ils s'aperçoivent que cette prétendue extraction des connaissances consiste en fait en un véritable processus constructif de modélisation.

Très vite, on a repris une hypothèse que Newell proposait dès 1982 sous le nom *d'Hypothèse du niveau des connaissances*, pour l'étendre à une véritable définition générale de la connaissance vue par les informaticiens. La connaissance devint ce dont a besoin un agent rationnel (humain ou artificiel) pour réaliser le Principe de rationalité suivant : dis-moi ce que veux, ce que peux, ce que sait, je te dirai ce que fais.

Autrement dit, la connaissance est devenu le lien essentiel entre les programmes et leur manifestation dans le monde, la connaissance faisant lien dualiste entre d'une part le code informatique interprété par les machines, et d'autre part l'agent artificiel émergent susceptible d'être engagé dans un processus de résolution collective de problèmes, mêlant des agents artificiels à des agents humains.

Nous pointerons les liens entre ce nouveau statut de la connaissance en informatique, qui culmine avec le WEB, et la simulation.

### L'informatique multimédia

Alors que l'intelligence artificielle opère essentiellement dans le champ de la langue, le multimédia voudrait opérer dans celui de la perception.

Mais comment communiquer l'impression perceptuelle que porte un mot comme "véhicule" sans réduire ce concept à une de ses représentations prototypales ? Au "tout langage" conceptuel de l'intelligence artificielle (manipulation de symboles interprétables en langue, programmation en langage, interfaces de contrôle d'inspiration langagière), le multimédia peut-il opposer une sorte de "tout gestuel" perceptuel et corporel ?

Pour beaucoup, la réponse est d'abord technologique, et y répondre consiste à disposer d'interfaces de contrôle de la perception (un peu à la manière des appareils de type réflexe dans le domaine de la photographie) et à se doter de moyens de modélisation et de rationalisation d'objets perceptuels permettant de les manipuler.

Mais dans la pratique, les applications multimédia réussies sont constituées d'astucieuses associations de données, de programmes de manipulation et d'interfaces, qui confèrent à un système artificiel une aptitude circonstancielle à être, non pas investi d'esprit ni même de corporéité, mais plutôt de contextualité.

L'informatique multimédia vise des outils qui permettraient de replier dynamiquement l'intention d'action sur la perception. Cette discipline précise ainsi la notion de simulation, dans un sens fort. Son enjeu semble bien être la réconciliation entre la perception et l'action, deux catégories séparées par la philosophie des Lumières.

C'est ainsi à une véritable déconstruction d'un divorce que nous invite l'informatique multimédia, si tant est qu'on accepte d'élargir sa définition à la mesure qu'elle exige. Cela augure de belles heures de recherches pluridisciplinaires. En particulier, les sciences cognitives d'inspiration analytique pourraient bien y perdre progressivement l'hégémonie qu'elle manifestent actuellement dans ces domaines.

Nous évoquerons cette question au travers d'une expérience visant la réalisation d'un opéra interactif, actuellement conduite par Alain Bonardi à l'IRCAM et à l'Université Paris 8.

## 5° Conclusion

Et si Diderot s'était trompé en opposant la dyade jugement/pénétration à la dyade sensibilité/enthousiasme dans sa théorie de la simulation ? Quelles en seraient les conséquences sur les conditions de possibilité d'une simulation informatique ?

## Références bibliographiques

### Sources philosophiques et littéraires

Aristote. "Métaphysique", *Garnier Flammarion*, 1993.

Denis Diderot. "Paradoxe sur le comédien", *Larousse*, 1934.

Eugen Fink. "Le jeu comme symbole du monde", *Les éditions de minuit*, 1966.

Witold Gombrowicz. "Ferdydurke", *Gallimard*, 1966.

Martin Heidegger. "Le principe de raison", *Tel Gallimard*, n° 79, 1962.

Hans Jonas. "Le phénomène de la vie", *De Boeck Université*, 2001.

Friedrich Nietzsche. "La volonté de puissance", *Tel Gallimard*, n° 259, 1995.

Robert Pirsig. "Traité du Zen et de l'entretien des motocyclettes", *Seuil*, 1978.

Henry David Thoreau. "Walden", *Gallimard*, 1922.

Uexküll. "Mondes animaux et mondes humains", *Denoël*, 1956.

### Bibliographie informatique et sciences

Bruno Bachimont. "Des machines qui pensent aux machines qui donnent à penser", thèse de doctorat, 1998.

Alain Cardon. "L'informatique, science ou technologie ? Un formidable défi pour l'intelligence artificielle !", *Bulletin de l'AFIA*, n° 42, juillet 2000.

Hubert Dreyfus. "Intelligence artificielle - mythes et limites", *Flammarion*, 1984.

Paul Jorion. "Le mathématicien et sa magie : Théorème de Gödel et anthropologie des savoirs", *Cahiers STP Maison des Sciences de l'Homme*, à paraître, décembre 2000.

Jean Lassègue. "Turing", *Les belles Lettres*, 1998.

Anne Nicolle. "Science et technique, l'évolution des rôles", *actes des Journées de Rochebrune*, à paraître, 2001.

François Rastier. "Représentation ou interprétation, perspective herméneutique sur la médiation sémiotique", *Presses universitaires de Grenoble*, 1996.

Jean-Michel Salanskis. “Le temps du sens”, *Editions HYG*, 1997.

Pierre Saurel. “Nécessité des modèles en sciences cognitives : de la modélisation à la mise en parangon”, thèse de doctorat en épistémologie, janvier 1998.

Didier Vaudène. “Une contribution à l'étude des fondements de l'informatique”, thèse de doctorat d'État, octobre 1992.

Terry Winograd et Fernando Flores. “l'Intelligence artificielle en question”, *PUF*, 1989.