INTELLIGENCE ARTIFICIELLE ET SYSTÈMES À BASE DE CONNAISSANCES

par Gilbert Paquette

SOMMAIRE

Introduction **1**
1. Intelligence artificielle, d'hier à aujourd'hui **1**
1.1 Définitions, objectifs et hypothèses de base 1
1.2 Étapes du développement de l'intelligence artificielle 2
1.3 Programmes qui imitent l'intelligence. 6
1.4 Intelligence naturelle et intelligence artificielle 10
2. Connaissances et inférence **11**
2.1 Représentation des connaissances 11
2.2 Programmes capables d'inférence 15
2.3 Inférence et résolution de problèmes 18
3. Systèmes experts et systèmes à base de connaissances. **20**
3.1 Expertise humaine et logicielle 20
3.2 Fonctionnement des systèmes à base de connaissances 24
3.3 Modes de développement 25
3.4 Domaines d'application 25
Conclusion **27**
À retenir **28**

Introduction

L'expression **intelligence artificielle** est encore entourée d'un certain mysticisme. L'idée de machine intelligente est de nature à faire peur à certains ou à être considérée comme une impossibilité par d'autres.

Pourtant, dès la naissance des premiers ordinateurs, on a assisté à la création de machines reproduisant un nombre de plus en plus grand de nos facultés intellectuelles, sans que nous nous en trouvions plus mal. Au contraire, ce sont les tâches intellectuelles les plus répétitives et les plus mécaniques qui ont été prises en charge par les machines dans la vie de tous les jours : calcul de longues séries de chiffres, recherche de livres dans une bibliothèque, mise à jour de comptes bancaires, révision d'un texte, etc. Grâce à l'ordinateur, nous sommes libérés d'un travail fastidieux et nous pouvons nous concentrer sur des questions plus importantes comme : quels calculs faire? quelles transactions effectuer? quoi écrire?

Il faut donc voir l'intelligence artificielle en continuité avec l'essor de l'informatique. Cependant, comme nous allons le constater, cette partie émergente de l'informatique s'attaque maintenant à des tâches intellectuelles de plus haut niveau.

Particulièrement depuis l'an 2000, l'intelligence artificielle ont quitté les laboratoires pour faire leur apparition à l'usine, au bureau ou même à la maison : programmes qui jouent aux échecs comme des maîtres, interfaces d'interrogation de banques de données en langue naturelle, programmes de traduction automatique, systèmes experts, pour aider à la prise de décision financière ou pour choisir un traitement médical, outils de formation « intelligents » et robots dirigés par logiciel et qui effectuent diverses tâches dans les usines.

Qu'y a-t-il de commun à toutes ces applications? Qu'est-ce qui distingue l'intelligence artificielle de l'informatique classique? Qu'est-ce qui nous distingue de la machine intelligente? En quoi les systèmes à base de connaissances ou les systèmes experts peuvent-ils être qualifiés « d'intelligents »? Voilà des questions auxquelles nous tentons maintenant d'apporter un début de réponse.

1. Intelligence artificielle, d'hier à aujourd'hui

On peut décrire ce qu'est l'intelligence artificielle par son évolution et ses champs d'application. On peut aussi en donner une définition, en identifier les grands objectifs et en décrire les hypothèses de base.

1.1 Définitions, objectifs et hypothèses de base

Définir l'intelligence artificielle n'est pas une tâche facile. Aussi peut-on retrouver plusieurs définitions dans les très nombreux livres et articles qui ont été écrits sur le sujet. En voici quelques-unes :

- L'intelligence artificielle est l'étude des facultés mentales à travers l'utilisation de modèles informatiques. (Chamiak et McDermott)
- L'intelligence artificielle est la discipline visant à comprendre la nature de l'intelligence en construisant des programmes d'ordinateur imitant l'intelligence humaine. (Alain Bonnet)

- L'intelligence artificielle est l'étude des idées qui permettent à des ordinateurs d'être intelligents (dans le but, d'une part, de rendre ceux-ci plus utiles et, d'autre part, de comprendre les principes qui rendent l'intelligence possible). (Patrick H. Winston)
- L'intelligence artificielle est l'étude des méthodes permettant aux ordinateurs de réaliser des choses pour lesquelles, en ce moment, les êtres humains sont supérieurs. (Elaine Rich)
- L'objet de l'intelligence artificielle est de reconstituer à l'aide de moyens artificiels – presque toujours des ordinateurs – des raisonnements et des actions intelligentes. (J. L. Laurière)
- L'intelligence artificielle est une science dont le but est de faire faire par une machine tout ce que l'homme est capable de faire comme par exemple jouer aux échecs, traduire un texte. Elle s'est donnée pour objectif d'analyser les comportements humains dans les domaines de la compréhension, de la perception, de la résolution des problèmes afin de pouvoir ensuite les reproduire à l'aide d'une machine. (Jacques Pitrat)

On peut ranger ces définitions en deux· grandes tendances. La première voit l'IA comme « l'étude des facultés mentales » et « la compréhension de la nature de l'intelligence en construisant des programmes imitant l'intelligence humaine ». La deuxième, au contraire, insiste davantage sur le caractère utilitaire de l'IA : « faire faire à la machine ce que l'homme est capable de faire », « rendre les ordinateurs plus utiles ». Les définitions de Laurière et de Winston sont plutôt neutres face à ces deux tendances : « doter la machine de comportements intelligents ».

On a longtemps opposé ces deux tendances. Aujourd'hui, la majorité des gens reconnaissent la nécessité d'en faire plutôt la synthèse. Pour rendre les ordinateurs plus utiles, il faut les doter de comportements intelligents, c'est-à-dire s'inspirer du fonctionnement de l'intelligence humaine. Inversement, pour comprendre l'intelligence humaine, on peut concevoir des programmes informatiques permettant de vérifier les théories psychologiques qui tentent d'expliquer notre fonctionnement cognitif.

On peut faire une analogie avec le rôle qu'a joué l'étude du vol des oiseaux dans le développement de l'avionique. Évidemment, personne ne prétendra que l'avion « vole » de la même façon que l'oiseau, mais en analysant par exemple la mécanique de l'aile des oiseaux, on a non seulement déterminé certains principes de conception d'une aile d'avion, mais on a aussi mieux compris comment un oiseau arrivait à voler.

D'une façon analogue, pour que l'ordinateur comprenne le langage naturel, reconnaisse une image, conseille un expert ou joue aux échecs, il faut d'abord s'interroger sur la façon dont l'intelligence humaine s'acquitte de ces tâches, sur ce qui fait leur difficulté et sur les processus de base qui permettent de les accomplir.

Au-delà de cet objectif de rendre l'ordinateur plus utile en prolongeant l'intelligence humaine, un autre objectif, indissociable du premier, est de mieux comprendre l'intelligence humaine. Implicitement, on fait l'hypothèse suivante : *les facultés intellectuelles peuvent être modélisées sous la forme des systèmes de symboles, les uns représentant les connaissances que l'on transforme, et les autres représentant les méthodes de transformation*.

1.2 Étapes du développement de l'intelligence artificielle

Avec la naissance de l'informatique, très tôt au début des années 50, on dépassera le stade des calculs numériques pour s'attaquer aux calculs formels où l'on manipule des symboles.

Ces programmes permettront à l'ordinateur de dépasser le stade de l'arithmétique et d'atteindre celui de l'algèbre et de la logique.

La naissance d'une discipline.

Si on doit apposer une date de naissance sur l'intelligence artificielle, il faut choisir 1956. Cette année-là, le Logic Theorist de Newell, Shaw et Simon est capable de démontrer des théorèmes de la logique des propositions. Lors d'une réunion regroupant une douzaine de chercheurs en cybernétique au Collège de Dartmouth la même année, ce programme les convainc des possibilités de manipulation de symboles de l'ordinateur. Le terme *artificial* intelligence est proposé et accepté pour désigner la nouvelle science. Les principaux participants à cette réunion, Allen Newell, John McCarthy, Marvin Minsky et Herbert Simon, deviennent ainsi les « pères » d'une nouvelle discipline.

À partir de 1956, les travaux progressent rapidement : jeux d'échecs, démonstration de théorèmes en géométrie, traduction automatique, etc. Les travaux autour du Logic Theorist mènent au General Problem Solver (GPS) capable de résoudre des problèmes dans divers domaines. On assiste à la création de langages de programmation adaptés à la manipulation de symboles sous forme de listes. Le langage LISP, le plus utilisé en intelligence artificielle, est créé en 1960.

Au début, deux approches sont utilisées. La première mise sur la puissance de calcul de l'ordinateur pour examiner le maximum de possibilités jusqu'à ce qu'une solution soit atteinte. La seconde s'inspire davantage du comportement humain en utilisant un ensemble de connaissances et des méthodes générales permettant d'éviter de considérer toutes les possibilités. Jusqu'en 1970, c'est la première approche qui sera nettement privilégiée car la puissance de calcul des ordinateurs croît rapidement et laisse croire que tout pourra être fait en augmentant simplement le nombre d'opérations à faire. De 1965 à 1970, on commence toutefois à s'attaquer à des problèmes où la puissance de l'ordinateur ne suffit plus. Dans des secteurs comme les jeux d'échecs, la traduction automatique ou la robotique, on se heurte à l'explosion combinatoire : le nombre d'opérations augmente avec la difficulté du problème de façon exponentielle.

Les premiers systèmes à base de connaissances

Au début des années 70, un mot d'ordre s'impose : contourner l'explosion combinatoire par l'observation plus attentive de l'être humain qui, après tout, réussit à accomplir toutes ces tâches sans une masse énorme de calculs et d'opérations.

Dans cet ordre d'idée, une équipe de l'Université Stanford développe un programme appelé DENDRAL pour analyser les résultats obtenus par un spectrographe de masse. La quantité de connaissances nécessaires pour analyser un spectre est énorme, mal connue et non répertoriée. Il faut donc l'extraire des experts humains et imiter leur comportement dans la sélection et le traitement des informations pertinentes. Alors que les premiers programmes d'intelligence artificielle étaient d'ordre général, DENDRAL fait appel à une importante base de connaissances spécialisées dans laquelle il choisit celles qui sont susceptibles de vraiment orienter les choix du système.

Pendant que l'on met ainsi au point les principes de base des systèmes experts, Terry Winograd du Massachusetts Institute of Technology (MIT) s'attaque à un tout autre domaine, mais avec une approche similaire. Son programme SHRDLU allait constituer une percée importante. Darts un micro-monde composé de blocs de différentes formes et de différentes couleurs, un programme parvenait à comprendre le sens d'une phrase au sujet de ces objets

et à utiliser cette compréhension pour se définir un plan d'action permettant de réaliser diverses manipulations. Grâce à cette approche axée sur des connaissances spécialisées, on assistera, entre 1970 et 1975, à de nouvelles percées en traduction automatique et en compréhension du langage naturel, au développement de nombreux systèmes experts, à des progrès marqués en reconnaissance des formes, au développement de la programmation logique et en représentation des connaissances.

La consolidation d'une discipline

Entre 1980 et l'an 2000, l'intelligence artificielle quitte les laboratoires de recherche pour s'introduire dans un grand nombre de secteurs de l'activité humaine. Le domaine des **systèmes experts** prend son essor. La plupart des grandes entreprises se donnent un ou plusieurs projets de développement de ces logiciels « intelligents » et certains d'entre eux entrent en opération. Ces logiciels « raisonnent » d'une façon calquée sur les raisonnements des spécialistes auprès desquels ont été recueillies les connaissances qu'ils exploitent dans des domaines aussi divers que la médecine, l'analyse financière ou la configuration d'appareils. Ils permettent notamment de simuler le comportement d'un expert dans des tâches de diagnostic, de surveillance ou de dépannage dans des installations industrielles coûteuses.

Un autre vaste champ d'applications est celui de la **robotique** et de la fabrication assistée par ordinateur (FAO). Depuis plusieurs années déjà, l'industrie utilise des milliers de machines programmables pour la réalisation de soudures par points, la pulvérisation de peinture, le chargement et le déchargement de machines et certaines opérations d'assemblage. L'objectif est de remplacer l'être humain dans des environnements hostiles ou des tâches fastidieuses. La plupart de ces robots sont affectés à des opérations répétitives dans un univers de travail bien défini et restreint. Cependant, on commence à introduire des robots dotés de facultés décisionnelles simples. Par l'intermédiaire de caméras ou de capteurs, ils acquièrent l'information nécessaire pour se mouvoir dans des environnements diversifiés.

Dans la **compréhension du langage** et dans la **traduction automatique**, des programmes agissent comme assistants pour la traduction de textes dans des domaines spécialisés. En littérature, des programmes d'analyse automatique de texte aident à analyser le contenu du texte et le style d'un auteur. De nombreuses interfaces en langage naturel permettent d'interroger une base de données en anglais ou en français. Ce développement est particulièrement important pour faciliter l'accès à l'informatique à un plus grand nombre de personnes.

En ce qui concerne la **reconnaissance de la parole**, des logiciels peuvent reconnaître les mots et les phrases et les écrire à l'écran. On peut commander l'ordinateur directement en prononçant des mots jusque dans nos automobiles et dans la commande de nos téléphones, nous libérant de l'obligation de dactylographie, permettant également aux personnes handicapées un accès à l'outil informatique.

Les systèmes d'IA sur le Web sémantique

Avec le début du siècle, le terme « systèmes experts » tombe progressivement en désuétude au profit de celui de « systèmes à base de connaissances (SBC) ». D'une part, les SBC sont intégrés comme modules « embarqués » dans des logiciels de l'informatique classique. De grandes compagnies d'informatique comme Oracle ou SAP intègrent des fonctionnalités de SBC dans leurs suites de produits comme moyen de spécification et de traitement des processus organisationnels.

D'autre part, on assiste à l'utilisation des SBC comme systèmes d'aide à la décision par l'analyse de données à l'aide de règles à côté de méthodes statistiques d'analyse de données. Un autre développement récent consiste en une gamme d'outils de recommandation de produits opérant sur le Web. On les retrouve par exemple comme conseillers pour le choix de livres ou de vidéos dans les sites d'opérateurs commerciaux comme Amazon ou eBay.

Mais le développement le plus prometteur est celui du Web sémantique. Reprenant les mêmes concepts centraux de représentation des connaissances et d'inférence, le fondateur du Web et actuel directeur du W3C (World Wide Web Consortium), Tim Berners-Lee propose au début des années 2000 d'intégrer aux ressources disponibles sur le Web des informations sur les connaissances qui y sont traitées. En décrivant leur sémantique, au-delà de leur syntaxe, on pourrait traiter les informations du Web par des agents informatiques capables de faire des recherches plus intelligentes qu'auparavant parce que basées sur le sens, sur une représentation des connaissances derrière les mots. Cette évolution du Web sera qualifiée de « Web sémantique ».

Les systèmes à base de connaissance de première génération étaient centralisés, demandant à chaque utilisateur de partager une même définition précise et rigide des concepts. De plus, les questions traitées par les systèmes experts étaient limitées de façon à ce que le système puisse répondre de façon fiable à un petit nombre de questions spécialisées. Au contraire, pour construire le Web sémantique, il faut maintenant accepter une décentralisation des concepts, une multiplicité des terminologies et un ensemble de problèmes à traiter pouvant évoluer dans le temps avec flexibilité.

Contrairement aux premiers systèmes d'intelligence artificielle (systèmes experts, analyse de textes, reconnaissance de formes), les systèmes du Web sémantique ne sont pas fermés sur une spécialité restreinte utilisant un vocabulaire fixe. Tout le monde peut dire n'importe quoi sur n'importe quel sujet. Le Web sémantique doit donc permettre de traiter une variété de concepts et de terminologies qui évoluent régulièrement, alimentés par un grand nombre d'intervenants sur le Web.

Pour ce faire, le W3C adopte un modèle de base de connaissances sous forme de graphe RDF (« Ressource development framework »). Un tel graphe est formé de triples sujet-propriété-valeur/objet qui permettent de décrire non seulement les propriétés des documents, des personnes et des organisations, mais aussi les concepts et leurs relations. Ces objets sont identifiés par leur URI. Chaque lien du graphe représente une propriété reliant un objet à un autre objet ou à une valeur (un nombre, un sujet, une date, …).

Cette méthode de représentations des connaissances permet de répondre à des requêtes intégrant des termes et des objets appartenant à plusieurs sources de données, par exemple « je cherche des ressources en physique, produites par la Télé-université et utilisées dans au moins un cours dans la francophonie ».

L'Ontology Web Language (OWL) fournit un langage universel de représentation des connaissances sur la base du RDF. Un langage de requête, le SPARQL, permet d'interroger la base de connaissances ou de lui ajouter de nouveaux triples. Des langages de règles comme le SWRL (« Semantic Web Rule Language ») sont aussi utilisés pour le traitement des connaissances.

Ainsi, l'intelligence artificielle et les systèmes à base de connaissances retrouvent une place centrale dans un nombre croissant d'applications actuelles du traitement de l'information sur le Web sémantique.

Le tableau 1 résume à grands traits l'évolution historique de l'Intelligence artificielle et des systèmes à base de connaissances que nous venons de présenter.

Tableau 1 – Évolution de l'IA et des systèmes à base de connaissances

1945	Premiers ordinateurs, début de l'information
1956	Naissance de l'intelligence artificielle
1960	Création du langage LISP
1968	DENDRAL, le premier système expert
1972	Création du langage Prolog pour la programmation logique
1980	Début de la programmation orientée-objet Départ de la micro-informatique
1985	Début d'une large diffusion des SBC et des systèmes d'IA
1991	Naissance du Word Wide Web
2001	Début du Web sémantique, l'IA et les SBC sur le Web

1.3 Programmes qui imitent l'intelligence.

Lors d'une table ronde tenue en 1985, Allen Newell, l'un des fondateurs de l'intelligence artificielle, en définit ainsi les bases : « L'intelligence artificielle est-elle cohérente aujourd'hui? Je le crois. Sa base qui repose sur les systèmes de représentation au moyen de symboles, l'utilisation universelle des arbres hiérarchiques de buts, l'influence des concepts de recherche et de connaissance et la façon dont ils s'articulent ensemble, les racines opérationnelles en logique classique et en démonstration de théorèmes, l'éventail des méthodes de résolution de problèmes et d'organisation de la mémoire, tout cela assure une cohérence suffisante pour l'avenir. »

On retrouve ici l'hypothèse de base que l'intelligence peut être reproduite grâce à des **systèmes de représentation symboliques**.

Prenons un exemple pour clarifier ces idées. Le taquin à 8 tuiles est un casse-tête formé de tuiles numérotées de 1 à 8 que l'on peut déplacer dans une grille 3 x 3 comportant une case vide. Partant d'une situation initiale où les tuiles sont en désordre, le but du jeu est de les replacer en cercle dans l'ordre suivant : 1, 2 et 3 sur la première ligne, 8, case vide et 4 sur la deuxième et 7, 6 et 5 sur la troisième.

Ayant décrit l'état final et l'état initial donnés dans l'énoncé du problème, il reste à constater à la figure 1 qu'il y a quatre opérateurs possibles :

H : déplacer la tuile du haut dans la case vide;
G : déplacer la tuile de gauche dans la case vide;
D : déplacer la tuile de droite dans la case vide;
B : déplacer la tuile du bas dans la case vide.

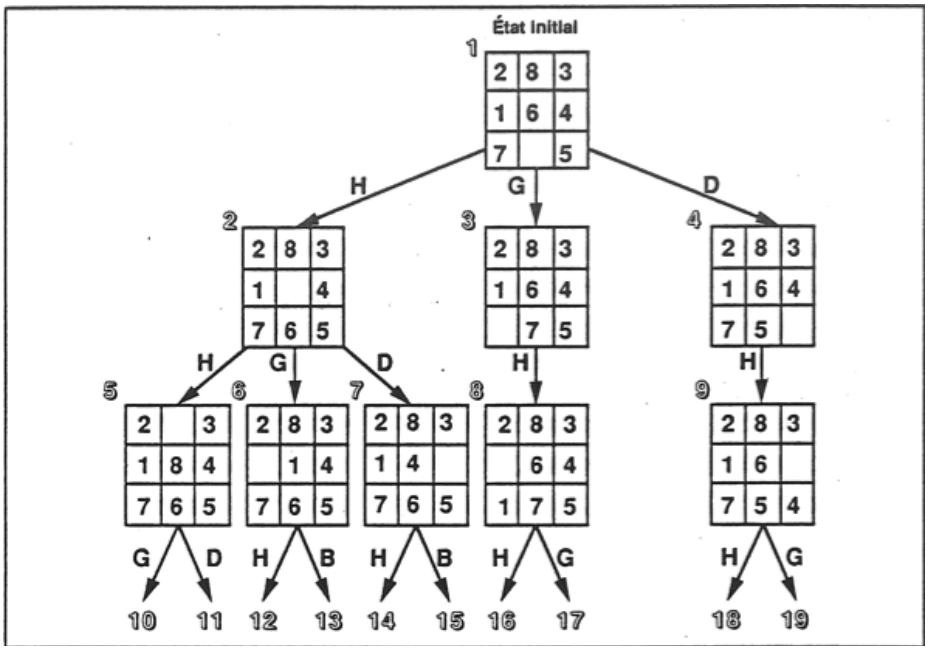

Figure 1 – Une démarche de résolution du problème du taquin à 8 tuiles

Une approche mécanique

Une façon classique de programmer la solution de ce problème consiste :

- à appliquer à l'état initial tous les opérateurs dans l'ordre H, G, D, B;
- à appliquer à chacun des quatre états à leur tour tous les opérateurs qui ne produisent pas une configuration déjà rencontrée;
- à procéder ainsi jusqu'à ce qu'une solution ait été atteinte ou que tous les chemins possibles aient été générés.

Cette approche mécanique a l'avantage d'être systématique. Toutefois, dès que le problème est quelque peu difficile, il faut abandonner l'idée d'une énumération exhaustive de tous les chemins.

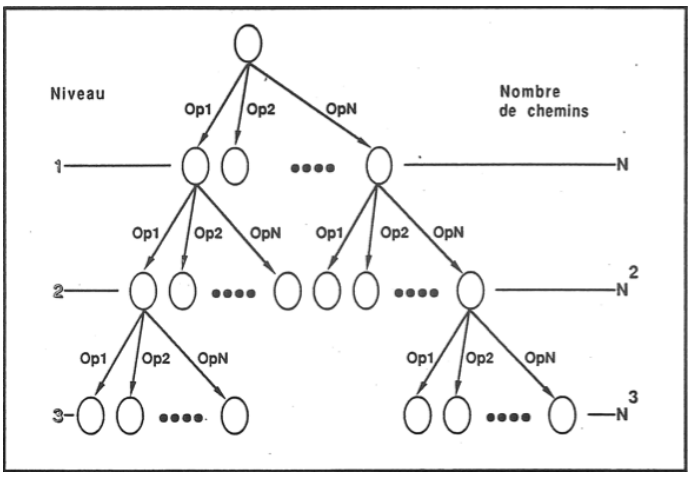

Figure 2 – L'explosion combinatoire du nombre de chemin

Règle générale, si à un état, on peut appliquer *N* opérateurs, après *a* niveaux, on aura généré un nombre de chemins égal à *Na.* Voilà très précisément le problème de *l'explosion combinatoire* ou *exponentielle* évoqué précédemment et comme le montre la figure 2.

Par exemple, au début d'une partie d'échecs, s'il y a 30 coups possibles en moyenne dans chaque position, après 10 coups joués, il y aura 30^{10} = 590 490 000 000 000 chemins qui auront été générés! Voilà qui peut excéder rapidement la capacité même d'un ordinateur très puissant.

Il faut alors trouver des moyens d'orienter la recherche dans des directions fructueuses, en augmentant « l'intelligence » du programme.

Une approche heuristique

Une façon d'augmenter l'intelligence d'un programme est de le doter d'une méthode d'évaluation des états qui associe à chacun une mesure de sa probabilité de conduire vers une solution, un peu comme le ferait un joueur humain. Pour chaque état, le programme examinera les valeurs des états successeurs obtenus en appliquant tous les opérateurs possibles et choisira de développer uniquement l'état ayant la meilleure valeur.

Une telle méthode est basée sur le raisonnement suivant : « moins il y a de tuiles mal placées après l'application de l'opérateur, plus je suis sur la bonne voie; par contre, si j'ai joué beaucoup de coups depuis le début, je dois avoir fait fausse route ».

Sur la figure 3, on a donné des numéros aux différents états E et on calcule pour chacun la fonction d'évaluation heuristique suivante :

f(E) = Nombre de coups depuis l'état initial + Nombre de tuiles mal placées.

Voyons de plus près comment on utilise une telle fonction. Dans la figure 3 nous avons utilisé *l'algorithme A** qui a été élaboré en 1968 par Hart, Nilsson et Raphael. On considère cet algorithme comme une méthode générale pour faire une recherche dans un arbre des possibilités. Rendu à un état quelconque dans l'arbre, on examine tous les opérateurs applicables et on choisit de n'appliquer que celui qui produira la plus petite valeur de la fonction d'évaluation.

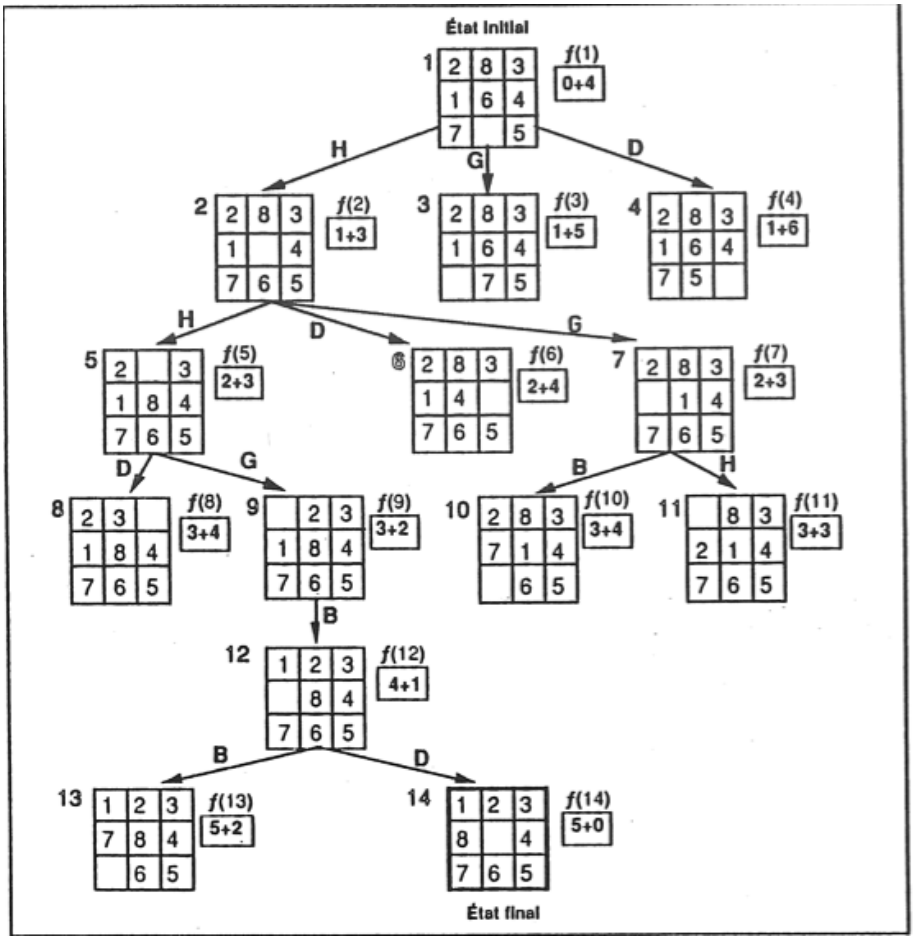

Figure 3 – Un exemple de fonction heuristique

Cette méthode est efficace dans la mesure où la fonction d'évaluation mesure bien la valeur de chacun des états. Dans le problème du taquin, elle va presque droit au but et trouve une solution en développant seulement 12 états intermédiaires.

Caractéristiques des méthodes d'IA

L'exemple qui précède met en évidence ce qui distingue les méthodes d'IA :

- elles cherchent à imiter les raisonnements humains plutôt que d'utiliser uniquement la puissance et la rapidité de l'ordinateur;
- elles traitent des représentations symboliques des situations plutôt que des nombres; par exemple, chaque situation du jeu sera représentée par une suite de symboles et les opérateurs, par les transformations qu'ils font à ces symboles;
- elles utilisent des méthodes de déduction plutôt que des algorithmes de calcul; par exemple, la fonction d'évaluation permettra au programme de déduire quel est l'opérateur le plus prometteur à utiliser;
- elles sont généralisables à des domaines souvent très différents; par exemple, le type de recherche qui précède peut être appliqué dans un système à base de connaissance qui doit faire un diagnostic pour trouver une pièce fautive dans un appareil ou pour faire le plan des déplacements d'un robot;

- elles utilisent des *heuristiques* ou méthodes informelles plutôt que des algorithmes rigides; par exemple, la deuxième méthode ci-dessus fait appel à deux heuristiques : le chemin doit être le plus court possible et la situation obtenue par un opérateur doit ressembler le plus possible à la solution finale.

Un autre exemple est fourni par les problèmes d'algèbre où un programme d'intelligence artificielle essaiera d'abord un algorithme pour simplifier l'équation, puis, si cela ne fonctionne pas, une décomposition en facteurs pour simplifier le problème, et au besoin une autre méthode totalement différente comme l'approximation successive de la solution. Rien ne garantit la découverte d'une solution, même s'il en existe une. Toutefois, si les heuristiques sont valables, le programme aura autant de chances de résoudre le problème qu'un être humain.

1.4 Intelligence naturelle et intelligence artificielle

On comprend mieux la difficulté des problèmes techniques que pose l'intelligence artificielle en comparant le cerveau humain aux ordinateurs modernes.

Les ordinateurs à base de silicium sont beaucoup plus rapides que les procédés électrochimiques à l'intérieur de nos cerveaux; la vitesse de propagation des impulsions électriques d'un ordinateur s'approche de celle de la lumière, c'est-à-dire 300 000 km par seconde, alors que les impulsions nerveuses se propagent à environ 500 km à l'heure, donc un million de fois moins rapidement. Il est évident que nos machines possèdent la vitesse nécessaire pour accomplir les processus de la pensée. Le problème ne se situe pas à ce niveau.

D'autre part, on estime que le cortex humain renfermant quelque dix milliards de neurones et que chacun de ceux-ci établit environ 10 000 synapses ou interconnexions avec les autres neurones. Conséquemment, le réseau neuronique du cerveau se compose de quelque 10^{14} de composantes actives. Du côté de l'ordinateur, on trouve couramment aujourd'hui des plaquettes de mémoire à semi-conducteurs (puces) renfermant des milliards composantes actives sur une surface de un centimètre carré.

Si les ordinateurs d'aujourd'hui sont plus rapides que nos cerveaux et occupent un espace parfois plus petit, on ne peut expliquer les différences de comportement que par l'architecture de la pensée qui nous distingue complètement des machines. Contrairement à l'ordinateur classique, le cours de nos pensées n'est pas séquentiel. Plusieurs processus parallèles concourent lorsque nous reconnaissons des formes, traduisons un texte ou résolvons un problème.

Cette hypothèse expliquerait notamment les faits suivants :

- Nous sommes lamentables en calcul arithmétique, même une calculatrice peut surpasser n'importe lequel d'entre nous, sans faire d'erreur en calculant.
- À l'exception de certains qui prétendent avoir une mémoire photographique, nous ne mémorisons pas rapidement l'information; une machine peut mémoriser parfaitement une énorme quantité d'informations textuelles.
- Nous confondons, associons, mélangeons les informations et nous les oublions sans avoir à en recevoir l'ordre; en fait, nous avons plus de difficulté à oublier quand on nous l'ordonne. (Essayez de ne pas penser aux éléphants pendant les prochaines 60 secondes.) À l'opposé, une machine n'oublie ou ne mélange jamais les informations à moins d'en avoir reçu l'ordre.

- Nous excellons dans les tâches de reconnaissance de formes et à repérer les informations utiles dans une masse de connaissances; la machine, par contre, n'arrive à le faire que plus difficilement.
- Certains de nos neurones meurent de temps à autre, sans que cela n'affecte notre aptitude à traiter l'information; au contraire, de légères défaillances dans le matériel suffisent à détruire l'aptitude d'un ordinateur à traiter l'information.

Nous semblons posséder une solide architecture qui manipule de façon médiocre les informations et les traitements séquentiels et systématiques, mais qui réalise des phénomènes stupéfiants avec d'énormes quantités d'informations qu'elle choisit soigneusement et traite en parallèle. Voilà pourquoi beaucoup pensent que les progrès les plus marquants de l'intelligence artificielle viendront du remplacement de l'architecture actuelle de nos ordinateurs par des réseaux de types neuronaux inspirés du fonctionnement en parallèle du cerveau humain.

2. Connaissances et inférence

En poursuivant cette comparaison avec l'intelligence humaine, on a subdivisé les travaux en intelligence artificielle en sous-domaines correspondant aux diverses facultés intellectuelles que l'on vise à reproduire artificiellement : représentation des connaissances en mémoire, méthodes d'inférence, résolution de problèmes, traitement du langage naturel, perception et reconnaissance des formes, induction et apprentissage.

Comme l'être humain, pour qu'un programme manifeste de l'intelligence dans un de ces domaines, il doit posséder en mémoire :

- une représentation des connaissances du domaine;
- des méthodes d'inférence permettant d'utiliser ces connaissances.

2.1 Représentation des connaissances

Pour échanger, transmettre ou traiter des connaissances, il est nécessaire de pouvoir les représenter sous une forme transmissible ou traitable. Le développement de systèmes de représentation adéquats pour la déduction, l'induction, la perception, l'apprentissage ou le traitement du langage est sans doute le domaine le plus fondamental en intelligence artificielle.

Examinons d'abord un exemple. Dans le manuel d'accompagnement d'un récepteur de télévision, les conseils sont parfois fournis comme dans la figure 4.

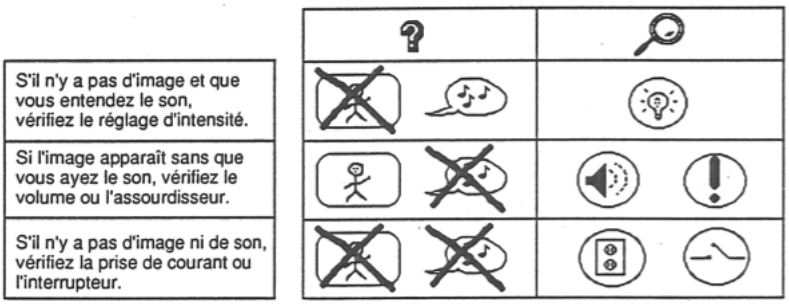

Figure 4 – Deux mode de représentation des mêmes connaissances.

Les textes et les images correspondantes de la figure 4 sont deux façons de représenter un certain nombre de *connaissances* utiles à la réparation de problèmes mineurs. La première représentation utilise la langue française alors que la seconde représentation ne contient aucun mot mais repose exclusivement sur l'interprétation des dessins.

Notion de représentation

Un ensemble de connaissances admettra en général plusieurs types de langages de représentations. *Comprendre* un texte ou un pictogramme consiste essentiellement à se construire une représentation mentale de la situation, c'est-à-dire un ensemble d'idées que l'on se forme sur la situation et que nous emmagasinons dans notre mémoire.

Grâce à ces représentations mentales, nous pouvons *communiquer* à quelqu'un les connaissances que nous avons acquises. Par exemple, si un problème se présente avec le téléviseur et que vous avez à l'expliquer à quelqu'un, vous n'utiliserez ni les mêmes mots ni les mêmes dessins que ceux du manuel, mais l'information sera quand même préservée. Cette représentation mentale préservée en mémoire peut être représentée par des systèmes de symboles que notre cerveau peut *traiter* pour créer de nouvelles représentations mentales. (Voir figure 5).

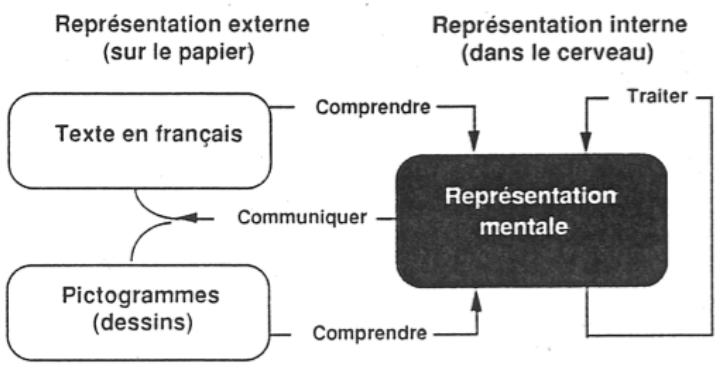

Figure 5 – Représentation externe et représentation mentale

Pour établir un système de représentation des connaissances intéressant pour la gestion des connaissances, nous aurons besoin de spécifier des conventions et des règles décrivant un langage symbolique. L'élaboration d'un système de représentation (un langage) comporte notamment les étapes suivantes.

1. *La définition d'une grammaire ou syntaxe décrivant l'ensemble des expressions acceptables obtenue en combinant les symboles de base d'un lexique.*

   Dans le cas de la langue française, le lexique est constitué des lettres de l'alphabet, des signes de ponctuation et des mots acceptés par un dictionnaire officiel. Dans le cas du langage musical, il s'agit de la portée et des symboles représentant les clefs, les notes, les silences, etc.

   Dans le cas de la langue française, la grammaire décrit comment composer l'ensemble des phrases du langage. Si nous voulons décrire la structure moléculaire de certains composés chimiques, l'ensemble des expressions acceptables consisterait en une suite de symboles chimiques avec des indices numériques comme H_2O ou NaCl. Dans le cas du langage musical, des règles de grammaire stipulent que chaque portée doit comporter une clef, un subdivision des temps, un respect de la subdivision en fonction de la valeurs des notes et des silences, etc.

2. *La définition d'une sémantique, soit une méthode pour donner un sens aux expressions de la grammaire.*

   Cela qui revient à associer aux expressions une représentation mentale intelligible et, inversement, une méthode pour associer une ou plusieurs expressions de la grammaire à des connaissances faisant partie du modèle mental de la personne qui s'exprime à l'aide du langage.

Un langage formel : la logique propositionnelle

Nous allons maintenant illustrer ce qui précède en élaborant un langage formel qui porte le nom de *logique propositionnelle*. Ce système permettra de représenter certains aspects de situations qui impliquent des raisonnements élémentaires, mais aussi de faire des requêtes dans une base de données.

Tableau 2 – Deux représentations d'énoncés simples

Langue naturelle	Langage schématique
S'il n'y a pas d'image mais que vous entendiez le son, vérifiez le réglage d'intensité de l'écran.	si non IMAGE et SON alors RÉGLAGE_INTENSITE
Si l'image apparaît sans qu'il y ait de son, vérifiez le contrôle du volume ou celui de l'assourdisseur.	si IMAGE et non SON alors VOLUME ou ASSOURDISSEUR
S'il n'y a ni image ni son, vérifiez la prise de secteur ou l'interrupteur principal.	si non IMAGE et non SON alors PRISE_SECTEUR ou INTERRUPTEUR
Mon appareil n'a ni son, ni image. La prise de secteur fonctionne pourtant bien. Où est le problème?	non SON et non IMAGE et non PRISE_SECTEUR

Le tableau 2 reprend les phrases concernant les problèmes de réception de la télévision. Il est facile de voir que les nouvelles expressions de la deuxième colonne traduisent l'essentiel de la situation par rapport au problème posé. Les « détails » comme les articles, les adjectifs ou les verbes (entendre, apparaître, vérifier) sont disparus. La structure française du texte a été tellement simplifiée que les nouvelles expressions ne sont plus acceptables au sens de la grammaire française, mais elles véhiculent l'essentiel de l'information qui nous intéresse.

La *grammaire* de la logique propositionnelle est constituée d'un lexique simple et de quelques règles permettant de combiner les termes du lexique et de définir les expressions correctes du langage.

- Le *lexique* de la logique propositionnelle se compose de deux types de symboles : soit les *propositions élémentaires* qui sont des suites de une ou plusieurs lettres majuscules telles que IMAGE, SON, ou PRISE_SECTEUR et les *connecteurs logiques non*, *et*, *ou*, *si… alors*.
- Les *règles de grammaire* du langage propositionnel stipulent le rôle de ces quatre connecteurs qui consiste à lier les propositions entre elles de façon à former des propositions plus complexes

- Le connecteur *non* peut être placé devant toute proposition simple ou composée entourée de parenthèses (comme le signe – devant un nombre)
- Les connecteurs *et*, *ou* et *si… alors* peuvent être placés entre deux propositions simples ou entre deux propositions composées entourées de parenthèses.

La sémantique de la logique propositionnelle consiste à interpréter chaque proposition élémentaire comme vraie ou fausse dans un contexte donné. Par ailleurs, les connecteurs ont un sens très précis qui est proche, mais non identique, au sens des mots correspondants de la langue française. Le connecteur si… alors exprime l'idée de conséquence. On appelle antécédent ou prémisse la première proposition qui y figure et conséquent ou conclusion la seconde. Elle est vérifiée si la conclusion est vraie dès que l'antécédent l'est. Une proposition contenant le connecteur ou est vérifiée si au moins une des deux propositions l'est. Une proposition contenant le connecteur et sera vérifiée si chacune des deux propositions l'est. Une proposition de la forme non sera vérifiée si la proposition qui suit le « non » est fausse.

Le caractère dépouillé et très structuré du langage propositionnel facilite certaines formes de raisonnement élémentaires basé sur les règles d'inférence suivantes :

- R1 : De (A) et de (Si A alors B), on peut déduire B (Si nous savons que deux expressions telles que « A » et « si A alors B » sont vraies, peu importe la signification accordée aux symboles A et B, on peut déduire que B est vrai). Cette règle a pour nom « modus ponens ».
- R2 : De (non A) et de (A ou B), on peut déduire B (D'une façon analogue, si nous savons que deux expression telles que « Non A » et « A ou B » sont vraies, peu importe la signification accordée aux symboles A et B, on peut déduire que B est vrai.)

Ce langage et ces règles d'inférence permettent de clarifier certains raisonnements. Reprenons les expressions du tableau 1 :

(1) si non (IMAGE et SON) alors RÉGLAGE_INTENSITE

(2) si (IMAGE et non SON) alors VOLUME ou ASSOURDISSEUR

(3) si (non IMAGE et non SON) alors (PRISE_SECTEUR ou INTERRUPTEUR)

(4) non IMAGE et non SON

(5) non PRISE_SECTEUR

et appliquons les règles d'inférence présentées plus haut. L'application de la règle R1 aux propositions (3) et (4) nous permet de déduire

(6) PRISE_SECTEUR ou INTERRUPTEUR (au moins l'une des deux est la cause). Puis l'application de la règle R2 à cette dernière proposition et à (5) nous permet de déduire qu'entre ces deux hypothèses, c'est l'interrupteur qui est défectueux.

Malgré sa simplicité, le langage propositionnel permet de démêler des situations beaucoup plus complexes que ce petit exemple, grâce à sa schématisation des structures de propositions et de la précision de ses règles de grammaire et d'interprétation. On retrouve ce type de langage à la base des systèmes experts : les propositions telles que (1), (2) et (3) constituant une base de connaissances et les faits énoncés en (4) et (5) constituant une base de faits. Dans un système expert, les règles de déduction telles que R1 et R2 constituent ce qu'on appelle : le « moteur d'inférence ». Avec un même moteur d'inférence, on peut en principe traiter des bases de connaissances dans des domaines très différents, en autant

qu'on puisse les représenter adéquatement dans un langage formel semblable à celui présenté ici.

2.2 Programmes capables d'inférence

Une fois l'ensemble des connaissances pertinentes à une situation traduit au moyen d'expressions, on obtient une *base de connaissances* dans le domaine choisi. Pour traiter l'information contenue dans la base de connaissances, nous avons besoin de « connaissances sur les connaissances » c'est-à-dire de connaissances d'ordre général qui indiquent comment accéder à un certain ensemble de connaissances, comment transformer cet ensemble ou l'augmenter. Dans l'exemple de la section précédente, nous avons rencontré ces connaissances générales sous la forme de deux règles d'inférence. Si nous voulons automatiser certaines facultés de raisonnement, il sera essentiel de transmettre ce type de connaissances à l'ordinateur.

L'idée de distinguer les connaissances spécialisées des règles agissant sur cet ensemble a permis plusieurs progrès importants dans certains domaines de l'intelligence artificielle. La technologie des systèmes experts a notamment profité de cette distinction en permettant le développement de moteurs d'inférence indépendants d'un domaine, les *coquilles (ou générateurs) de systèmes experts*. Ces logiciels contiennent une représentation des connaissances générales nécessaires au traitement de règles et de faits et permettent à un utilisateur de se concentrer uniquement sur la construction d'une base de connaissances spécialisées.

Un exemple d'inférence

Nous allons maintenant décrire comment il est possible de raisonner automatiquement avec les propositions, comme le ferait une machine. Cette description est basée sur la règle de déduction *modus ponens* qui s'énonce ainsi: si on a les deux propositions suivantes sont vérifiées : « si **A** alors **B** » et « **A** », on peut déduire que la proposition **B** est vérifiée.

Considérons une situation fictive où nous disposons d'une base de connaissances (faits et règles). Le lecteur peut imaginer que ces lettres représentent des propositions au sujet d'un meurtre dont on veut découvrir le coupable. Par exemple, A représente le fait « André est innocent »; B, « Bernard est innocent », etc.

Pour ce qui est des règles « si A alors B » signifie que si André est innocent alors Bernard est innocent.

Supposons maintenant que nous ayons les règles et les faits suivants :

1. si A alors B
2. si F alors G
3. si (G et H) alors J
4. si (B et C) alors F
5. si (G et K) alors L
6. si A alors Z
7. si W alors J
8. H
9. C
10. A

À l'aide de ces faits, nous tenterons d'établir le fait J (Jean est innocent). Nous pouvons consigner sur un bloc-notes les faits élémentaires qui sont vérifiés. Chaque fois qu'un nouveau fait est vérifié, on parcourt la liste des propositions de la forme « si… alors... » pour tester si l'antécédent est vérifié. Si c'est le cas et que le conséquent n'est pas dans notre liste, nous l'ajoutons. Lorsque nous obtenons J ou que nous ne pouvons plus rien ajouter, nous arrêtons.

- Dans l'exemple qui précède, la liste initiale des faits connus est : H, C, A
- L'antécédent de la proposition 1 étant vérifié, nous ajoutons B à notre bloc-notes pour obtenir H, C, A, B
- L'antécédent de la proposition 6 est vérifié, nous ajoutons Z : H, C, A, B, Z
- L'antécédent de la proposition 4 est maintenant vérifié, nous ajoutons F : H, C, A, B, Z, F
- Puis, l'antécédent de la proposition 2 est vérifié, nous ajoutons G : H, C, A, B, Z, F, G
- Enfin, l'antécédent de la proposition 3 est vérifié, nous ajoutons J : H, C, A, B, Z, F, G, J, ce qui termine les inférences puisque nous avons déduit que Jean est innocent.

Notons que nous n'aurions pas pu établir le fait « L » puisqu'à la dernière étape, il n'y avait plus aucune règle qui puisse s'appliquer. Cette technique est simple et facile à automatiser; mais elle génère parfois des résultats inutiles, comme le fait « Z » qui ne sera pas utilisé. C'est un exemple de méthode d'inférence bien adaptée à une représentation des connaissances sous forme de proposition.

Modes de représentation structurée

Il existe des modes de représentation autres que la logique propositionnelle et qui exigent des méthodes d'inférence tout à fait différentes, par exemple les représentations par schéma ou par objet. Celles-ci sont à la base de la représentation par graphe RDF à la base du Web sémantique. Plutôt que d'accumuler les énoncés comme en logique propositionnelle, on peut se concentrer sur les objets dont on parle dans les énoncés. On regroupe alors autour de chacun les propriétés que l'on connaît de cet objet.

Prenons l'exemple du règne animal. On pourra constituer une base de connaissances comme celle du tableau 3.

Tableau 3 – Un ensemble de schémas décrivant des animaux.

Objet	Propriété	Valeurs
Animal Vertébré	sorte-de Reproduction : Alimentation Locomotion Enveloppe Structure	Animal – – – – osseuse
Oiseau	sorte-de Reproduction Alimentation Locomotion Enveloppe Structure	Animal vertébré Ovipare – – plumes –

Objet	Propriété	Valeurs
Mammifère	sorte-de Reproduction Alimentation Locomotion Enveloppe Structure	Animal vertébré Vivipare – – poils –
Loup	sorte-de Reproduction Alimentation Locomotion Enveloppe Structure	Mammifère – carnivore membres – –

Une fois les objets définis de cette façon dans la base de connaissances, on a une hiérarchie naturelle créée par la relation « sorte-de ». Le graphe de la figure 6 représente cette hiérarchie appelée aussi *taxonomie*.

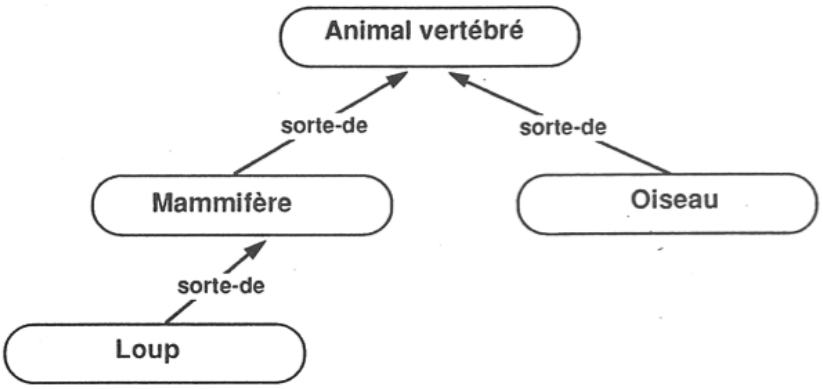

Figure 6 – Graphe RDF d'une taxonomie de concepts.

Ainsi cette sorte de relation nous permet de définir tout nouvel objet par ses différences avec ceux déjà définis. Par exemple, si on veut définir l'objet « loup », on n'a qu'à spécifier son parent « mammifère » et ce qui le distingue d'autres mammifères : il est carnivore et se déplace au moyen de membres.

Si on veut savoir quel est le mode de reproduction du loup, on regarde si l'attribut correspondant à une valeur. Comme ce n'est pas le cas, on remonte voir son parent « mammifère » et on trouve *vivipare.* Par contre, si on veut connaître la structure du loup, on ne peut la trouver dans l'objet « mammifère ». On remonte donc jusqu'à son grand parent « animal vertébré » et on trouve : osseuse.

Le mécanisme que nous venons de décrire est un mécanisme d'inférence bien adapté à la représentation par objets. On l'appelle héritage de propriétés. Lorsqu'on ajoute un nouvel objet à la base de connaissances, on n'a qu'à dire quel est son parent et ensuite quelles propriétés le distinguent de ce dernier. Le mécanisme d'héritage nous assure que toutes les propriétés des ancêtres seront transmises à ce nouveau descendant.

En règle générale, les représentations par objets seront plus complexes que dans cet exemple. On y retrouvera, par exemple, l'héritage multiple, chaque nouvel objet pouvant hériter de plus d'un parent. Il faudra alors résoudre les conflits possibles. D'autre part, chaque valeur d'une propriété d'un objet peut être calculée par une procédure. On peut aussi donner une valeur implicite à certaines propriétés qui s'appliquera tant qu'on ne lui aura pas donné explicitement une valeur différente. Il faudra alors spécifier quelle valeur aura la préséance : la valeur implicite, la valeur héritée ou la valeur calculée.

Comme on peut le constater, il y a plusieurs méthodes d'inférence possibles non seulement quant à la forme que prendra le mécanisme d'héritage, mats aussi quant à la façon dont les divers objets seront sollicités pour accomplir une tâche.

2.3 Inférence et résolution de problèmes

Résoudre des problèmes est une des fonctions fondamentales de l'intelligence humaine qui exige, elle aussi, une bonne représentation des connaissances et des méthodes d'inférence dont l'application permettra de trouver la solution. Poser un problème, c'est d'abord en comprendre l'énoncé. Or, le langage naturel a les défauts de ses qualités : il est la plupart du temps incomplet et ambigu et parfois redondant. Les détails qui manquent sont souvent implicites et doivent être déduits en faisant appel à des connaissances générales que possède tout être humain. La redondance sert à souligner par la répétition les éléments importants. La première chose à faire pour résoudre un problème est donc d'en décoder le sens, de trouver une représentation des connaissances où l'essentiel est dit.

On construit ainsi une base de connaissances qui, au départ, contiendra une description de l'état initial. La base de connaissances contiendra également des opérations exprimées par des règles permettant de transformer cet état initial en d'autres états qui, petit à petit, nous rapprocheront de l'état final ou solution.

Prenons le problème suivant : deux pots A et B contiennent respectivement 9 et 4 litres. Il n'y a aucune marque sur les pots. Peut-on obtenir exactement 3 litres dans le contenant A, uniquement en remplissant ou en vidant les pots ou en les versant l'un dans l'autre?

Si vous essayez d'abord de résoudre ce problème par vous-même, vous constaterez que l'on peut obtenir cinq litres dans A en deux étapes :

- Remplir A,
- Verser le contenu de A dans B jusqu'à remplir B.

On peut représenter les connaissances nécessaires pour résoudre ce problème à l'aide d'états et d'opérateurs comme suit :

- les *états* sont de la forme [A] [B] où A et B sont le nombre de litres de chaque récipient; par exemple, [4] [5] signifie que le récipient A contient 4 litres et le récipient B, 5 litres.
- les *opérateurs* sont représentés par les règles SI… ALORS de transformation entre états permises par l'énoncé du problème tel qu'indiqué au tableau suivant.

Opérateur	Si	Alors
Remplir A	[A] [B]	[9] [B]
Remplir B	[A] [B]	[A] [4]
Vider A	[A] [B]	[0] [B]

Vider B	[A] [B]	[A] [0]
Verser B dans A	[A] [B] et A + B < = 9	[A + B] [0]
Verser B dans A	[A] [B] et A + B > 9	[9] [A + B - 9]
Verser A dans B	[A] [B] et A + B < = 4	[0] [A + B]
Verser A dans B	[A] [B] et A + B > 4	[A + B - 4] [4]

La présence de deux opérateurs pour le versement tient compte du fait que le calcul sera différent si on peut verser complètement un récipient dans un autre ou seulement une partie. Par exemple, l'avant-dernier opérateur signifie que si le total des deux récipients est inférieur ou égal à 4 litres, on peut verser tout le contenu de A dans B, lequel contient alors A + B litres, alors que A est devenu vide. Dans le cas contraire, l'autre opérateur affirme que l'on pourra remplir B jusqu'à 4 litres et que dans A, on ajoutera B - 4 litres à A qui contiendra alors A + B - 4 litres.

Pour utiliser ces connaissances, le programme devra contenir un moteur d'inférence qui « instancie » les variables A et B c'est-à-dire leur donne des valeurs particulières correspondant à un état déjà obtenu. Après cette opération, le moteur peut appliquer la règle de *modus ponens* et utiliser ces opérateurs comme des énoncés en logique propositionnelle.

Par exemple, ce moteur d'inférence produira la solution du problème « obtenir 5 litres dans A » comme suit :

- [0] [0] (état initial, un fait)
- Si [0] [0] alors [9] [0] (**instanciation** de l'opérateur remplir A)
- [9] [0] (application de *modus panens* à (1) et (2))
- Si [9] [0] et 9 + 0 > 4 alors [5] [4] (instanciation de la dernière règle)
- [5] [4] (application de *modus panens* à (3) et (4))

Le problème se présente donc comme une suite d'états obtenue en appliquant une suite d'opérateurs (instanciés correctement) à partir de l'état initial jusqu'à l'état final soit [0] [0]; puis remplir A [9] [0]; puis vider A dans B [5] [4].

Le problème « obtenir 3 litres dans A » n'est pas aussi simple et nécessite d'essayer divers chemins en appliquant tous les opérateurs possibles à une situation avant de trouver un chemin-solution comme en témoigne la figure 7.

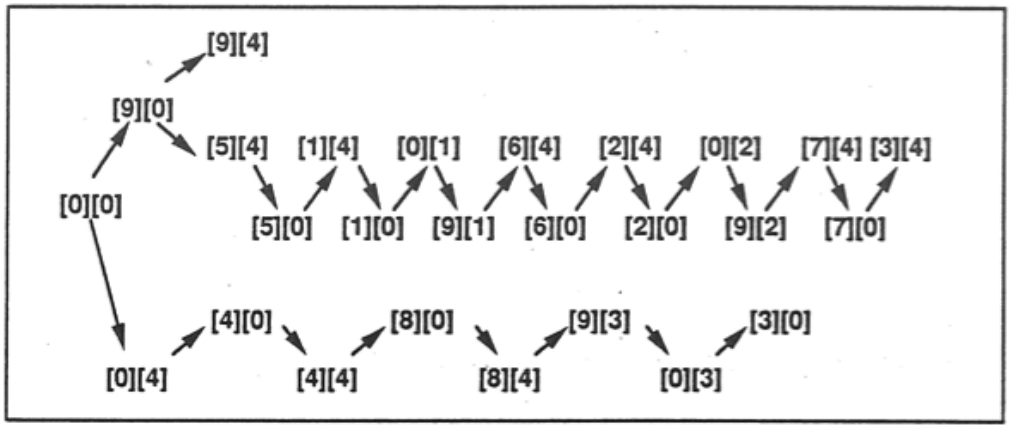

Figure 7 – Démarche de résolution du problème des récipients.

Cette démarche peut être faite par un ordinateur muni d'un moteur d'inférence capable d'instanciation de variables. À partir d'un fait initial [0] [0], on applique systématiquement les règles permises (les opérateurs) en déduisant toujours des faits nouveaux jusqu'à déduire la solution, un fait du type [3] [B].

On s'aperçoit qu'il y a plusieurs façons de parcourir l'arbre des possibilités. Notamment, l'ordre dans lequel on applique les opérateurs est critique. Ici, on perd beaucoup de temps à développer la « branche » [5] [4], alors que la « branche » [0] [4] conduit deux fois plus rapidement à une solution. L'étude des méthodes les plus efficaces de recherche dans un arbre des possibilités sert à la fois à trouver de bonnes méthodes de déduction ou d'inférence et de bonnes méthodes de résolution de problèmes.

L'inférence, tout comme la résolution d'un problème, sont tous deux des mécanismes de recherche dans un arbre des possibilités qui peuvent être décrits ainsi : en partant d'une situation initiale connue (les faits), atteindre une des situations finales (fait-conclusion, fait-solution) désirées en appliquant une suite finie d'opérateurs à la situation initiale ou aux situations déjà obtenues de celle-ci.

3. Systèmes experts et systèmes à base de connaissances.

Les systèmes à base de connaissances ont quitté les laboratoires de recherche pour faire leur entrée dans différents secteurs de l'activité humaine. On retrouve aujourd'hui, partout dans le monde, des systèmes experts qui œuvrent dans des secteurs aussi divers que la planification financière, l'analyse de prêts, le design en avionique, la configuration d'ordinateurs, le diagnostic de réseaux de communications ou de transports, le diagnostic médical, le forage des sols et la fabrication assistée.

3.1 Expertise humaine et logicielle

Qu'est-ce qu'un système expert? On peut en donner une première définition :

C'est un logiciel jouant le même rôle qu'un expert humain, c'est-à-dire capable, à partir de ses connaissances spécialisées, en appliquant des méthodes d'inférence, de fournir des conseils relatifs à son domaine d'expertise.

Pour bien comprendre cette définition, examinons ce que fait un expert humain dans un domaine donné. Prenons l'exemple du mécanicien auquel nous confions notre voiture. Il doit être capable de poser un diagnostic sur les problèmes et effectuer des réparations. Pour cela, il doit posséder de bonnes connaissances sur les différents sous-systèmes qui composent l'automobile et sur les pièces qui en font partie. Il doit également connaître des méthodes qui lui permettent de localiser la source de mauvais fonctionnement. Il doit enfin posséder d'autres connaissances sous forme de règles pour appliquer une méthode appropriée de réparation.

Dans un tout autre domaine, le conseiller financier doit être capable d'analyser la situation financière d'une famille par ses différents bilans et prévisions budgétaires. Il doit tenir compte aussi des investissements (maison, fonds de pension, obligations et actions), des dettes et de la situation fiscale des membres de la famille. Une fois qu'il trouve un déséquilibre, il devra suggérer une méthode de redressement : payer les dettes à court terme et à intérêt élevé, faire un emprunt pour investir dans tel ou tel secteur, etc.

À ces deux exemples, on pourrait ajouter ceux du médecin, de l'avocat ou du professeur dans le cas desquels on s'attend également à une forme semblable d'expertise. En résumé, pour être considéré comme un expert, on doit :

- Posséder un *savoir spécialisé* qui regroupe de vastes connaissances dans un domaine donné. Ce savoir ne se limite pas à des connaissances factuelles, mais il comprend un réseau de concepts, de règles, de méthodes qui forment l'expertise.
- Savoir *utiliser les connaissances efficacement.* Il ne suffit pas à l'expert de posséder de vastes connaissances, il doit pouvoir les utiliser rapidement et avec succès. Pour cela, il doit savoir poser les bonnes questions au client de façon à repérer les connaissances qui s'appliquent dans son cas.
- *Connaître ses limites* de façon à ne pas s'occuper de questions hors du cadre de sa compétence.
- *Fournir des explications appropriées sur les informations qu'il donne* ou les solutions qu'il propose. Généralement, ce n'est pas l'expert qui décide, mais celui qui demande conseil. Il est donc un conseiller qui doit être capable de convaincre le client que sa solution est bonne.

La consultation d'un expert est donc essentiellement un dialogue où celui-ci pose des questions de plus en plus précises pour identifier les connaissances dont il a besoin et où il explique les raisons de ses recommandations. Dans un tel dialogue, le client fournit des faits et pose des questions pour bien comprendre le sens de la solution proposée par l'expert. À la fin, il lui revient de prendre la décision finale.

Les premiers systèmes experts

Des dizaines de systèmes experts construits sur ce modèle ont été développés aux cours des années 70 et 80. En voici quelques exemples parmi les projets pionniers.

Le système DENDRAL de l'université Stanford a été le premier système expert à être opérationnel. Le projet a débuté au début des années 70. DENDRAL œuvre dans le domaine de la chimie. Il analyse des données d'expériences en spectrographie de masse, en résonance magnétique du noyau et autres pour inférer les structures possibles d'un composé chimique inconnu. Le système DENDRAL génère toutes les structures possibles en produisant systématiquement les structures partielles compatibles avec les données et en les élaborant de toutes les façons plausibles. En éliminant rapidement toutes les sous-structures non plausibles, il évite une recherche exponentielle. En produisant systématiquement toutes les structures plausibles, il trouve même des possibilités occasionnellement négligées par des experts humains. Le système DENDRAL, étant supérieur à n'importe quel humain dans la tâche limitée qu'il accomplit, a provoqué une redéfinition du rôle des chercheurs en chimie. Il est aujourd'hui utilisé couramment dans les laboratoires américains.

Une autre direction de développement a mené au système MACSYMA, un système expert sur les mathématiques symboliques mis au point à l'institut de technologie du Massachusetts. Tout comme DENDRAL, MACSYMA surpasse la plupart des experts humains. Il fait du calcul différentiel et intégral et excelle dans la simplification de pressions symboliques. Utilisé quotidiennement par des chercheurs en mathématiques et des physiciens dans le monde entier, MACSYMA intègre des centaines de règles recueillies chez des experts en mathématiques appliquées. Chaque règle donne une façon de transformer une expression en une autre expression équivalente; la solution d'un problème nécessite la découverte d'une chaîne de règles qui transforment l'expression originale en une expression convenablement simplifiée.

Le système CADUCEUS de l'université Carnegie-Mellon de Pittsburgh consiste en un très vaste réseau sémantique de relations entre des symptômes et des maladies qu'on retrouve en médecine interne. En 1982, le système détenait approximativement 100 000 associations représentant près de 85 % des connaissances pertinentes. De plus, il utilise quelques stratégies complexes qui cherchent à distinguer plusieurs maladies à partir des mêmes symptômes. Des tests effectués sur le terrain avec l'aide du National Institute of Health détermineront la pertinence de l'utilisation de CADUCEUS en situation réelle.

Le système MYCIN est un autre système expert qui a eu une influence considérable sur l'évolution du domaine depuis le début des années 70. Il s'attaque au problème du diagnostic et du traitement des maladies infectieuses du sang. Sa base de connaissances comprend à peu près 400 règles qui mettent en relations des états possibles et des interprétations correspondantes. Lorsqu'il résout un problème, MYCIN compare les conditions d'application d'une règle aux données disponibles ou encore, il demande des données additionnelles au médecin. Le cas échéant, il essaie de déduire la vérité ou la fausseté d'une condition à partir d'autres règles. Lorsqu'un groupe d'experts a évalué les performances de différents intervenants incluant des experts médicaux, des internes et MYCIN, les performances de MYCIN ont été jugées aussi bonnes ou supérieures à celles d'experts humains.

À Stanford, on a développé une version de MYCIN appelée EMYCIN qui en comprend tous les éléments à l'exception des connaissances sur les maladies du sang. EMYCIN a été le premier système essentiel ou coquille de système expert. Il comprend un moteur d'inférence, un mode de représentation des connaissances et des modules d'explication et de dialogue avec l'usager. En lui ajoutant une base de connaissances, on a construit d'autres systèmes dans d'autres domaines.

L'utilisation que fait MYCIN de règles « si… alors… » a favorisé l'apparition d'une gamme de systèmes semblables. Chez SRI, PROSPECTOR utilisait une forme similaire pour représenter des connaissances sur les relations entre des dépôts de minéraux. PROSPECTOR inclut maintenant près d'une douzaine de bases de connaissances pour divers types de dépôts. Comme MYCIN, il détermine le diagnostic le plus plausible en évaluant jusqu'à quel point chaque condition antécédente est appuyée par les faits. On applique ce procédé de façon récursive jusqu'à ce que les données sur toutes les conditions pertinentes aient été demandées et combinées selon une méthode heuristique.

Les systèmes à base de connaissances actuels

Plusieurs systèmes récents qui intègrent la technologie des systèmes à base de connaissances dans des systèmes plus larges de support aux entreprises. L'interface usager est souvent offerte par l'intermédiaire d'un fureteur sur le Web.

- ARCTOOLS (http://www.arctools.com/technology/oracle/) est un moteur d'inférence à base de règles qui peut être paramétrisé pour filtrer et archiver les bases de données Oracle. L'outil crée du code PL/SQL sur mesure en cours d'exécution pour optimiser la performance pour toutes les applications utilisant ORACLE. Il crée également une application d'archivage général. Les modules fonctionnels d'une application enregistrés dans des tables relationnelles sont interprétés à l'aide des règles intégrées dans ARCTOOLS et transformés en énoncés SQL par le moteur d'inférence d'ARCTOOLS. Un composant d'interface usager écrit en html/javascript/jsp/servlet donne accès à l'outil à partir d'un fureteur Web.

- Le système de gestion d'entreprise SAP, probablement le plus répandu, offre d'utiliser un moteur à base de règles pour traiter l'information des processus d'affaire (SAP Business Workflow). Les règles sont construites sur mesure pour déduire des informations à partir de la définition d'un « Workflow ». Pour construire des agents à base de règles, il faut a) définir quelles informations seront fournies à l'agent pour déclencher ses règles et b) fixer les paramètres de contrôle du moteur d'inférence. On peut ainsi, en cours d'exécution du Workflow, offrir aux participants des informations qui ne sont pas explicitement décrite dans l'interface du Workflow.

- Certains sites de vente en ligne comme AMAZON utilisent un système de recommandation qui offre une forme de filtrage de l'information présentant à un usager des livres, des films, des enregistrements musicaux susceptibles de l'intéresser. Ce type de système compare le profil d'un utilisateur (ce qu'il a commandé, ou les objets qu'il regarde dans un site, …) avec les caractéristiques des objets (leur contenu, les médias utilisés, …). Les systèmes de recommandation à base de règles gèrent le filtrage à l'aide de règles qui sont évaluées par un moteur d'inférence.

Il existe plusieurs système générateurs (ou « coquilles ») qui permettent de produire des SBC dans différents domaines. On en trouve une liste à l'adresse http://www.kbsc.com/rulebase.html, dont les suivants :

- EXSYS-CORVID est un outil de développement générique qui offre à l'usager des outils de consultation d'un système à base de connaissance construit par un concepteur et des outils permettant d'afficher, de modifier, d'imprimer une ou plusieurs bases de faits résultant de la consultation du système. En même temps, EXSYS offre au concepteur outils d'aide à la construction d'un système à base de connaissances, des outils d'édition automatique de l'interface usager, des outils facilitant l'intégration et la validation des connaissances. Créé initialement en 1983, le système a migré sur l'Internet au milieu des années 90 et il est régulièrement amélioré. C'est l'un des outils les plus largement utilisés pour capter et diffuser l'expertise sous forme de règles. Sur le site (http://www.exsyssoftware.com/CaseStudySelector/casestudies.html), on présente de nombreuses applications dans tous les secteurs de l'industrie.
- VANGUARD Knowledge Automation System (http://www.vanguardsw.com/products/knowledge-automation-system/) est un autre outil de développement de SBC, très utilisé notamment dans les systèmes d'appel et les processus d'affaire dans les organisations. Le système traduit de façon automatique des arbres de décision et des règles sous forme de dialogues dans un fureteur Web. Vanguard combine les systèmes à base de règles à des fonctionnalités d'analyse dans des domaines comme la prédiction, les échéanciers de production, la planification des ressources et les agents d'assistance aux usagers.
- JESS (http://herzberg.ca.sandia.gov/jess/index.shtml) est un moteur d'inférence à base de règles écrit entièrement en langage Java. Il permet d'écrire des logiciels qui ont la capacité de raisonner en utilisant des connaissances qu'on leur fournit sous forme de règles. Jess inclus un environnement de développement élaboré basé sur la plateforme Eclipse. Il fournit un langage de script qui peut appeler des méthodes Java et réaliser des interfaces sur Internet.

Dans le cadre du Web sémantique, les SBC sont développés en combinant ontologies, moteurs d'inférence et systèmes de règles. L'outil probablement le plus utilisé pour ce faire est PROTÉGÉ (http://protege.stanford.edu), utilisé par plus de 240 000 usagers. Ce système

permet de construire des ontologies selon le standard OWL en décrivant les concepts d'un domaine et leurs propriétés. Les instances d'une ontologie sont des triplets (objet-propriété-valeur) ou (objet-propriété-objet) où les objets sont membres des classes représentant les concepts. Ces triplets forment une base de connaissance dont les triplets respectant le standard RDF. PROTÉGÉ intègre des moteurs d'inférence (Pellet, Fact+) qui calculent les valeurs de propriétés d'un objet à partir d'autres valeurs déjà spécifiées. Dans PROTÉGÉ, on peut utiliser un module, SWRLTab qui permet de construire des règles exprimées à l'aide de la terminologie d'une ontologie OWL et du standard SWRL (Semantic Web Rule Language). Ce module soutient l'édition et l'exécution des règles dans PROTÉGÉ. Il permet l'extension à divers moteurs d'inférence, ainsi que l'interrogation des ontologies pour faire des recherches d'informations ou d'autres types de traitement.

3.2 Fonctionnement des systèmes à base de connaissances

En 1977, Feigenbaum, l'un des pionniers dans le domaine des SBC, émettait une observation qui résumait un changement majeur de perspective dans les travaux d'intelligence artificielle liés aux systèmes experts : « La puissance d'un système expert provient des connaissances qu'il possède plutôt que des formalismes particuliers ou des méthodes d'inférence qu'il utilise ».

Quels que soient les outils, leurs formalismes ou leurs méthodes d'inférence, les systèmes à base de connaissances, tout comme les systèmes experts, sont fondés sur une séparation nette entre les connaissances et les méthodes d'inférence. Ainsi les connaissances, au lieu d'être imbriquées dans la structure du programme (comme des commandes « if… then… else… » d'un langage de programmation), sont considérées comme des données interchangeables, susceptibles d'être corrigées, mises à jour et exploitées par des programmes (les moteurs d'inférence) qui, eux, simulent les mécanismes de raisonnement et assurent l'interface avec les utilisateurs.

Dans tout système à base de connaissance ou système expert, on retrouvera à la base les composantes suivantes :

- La *base de connaissances* est propre au domaine traité et, contrairement aux bases de données, n'est pas limitée aux connaissances de type factuel, aux données. Elle rassemble tous les types de connaissances pertinentes pour le domaine considéré : description des objets et de leurs relations (par exemple sous forme d'ontologie), règles à appliquer pour poser un diagnostic ou résoudre un problème, méta-connaissances permettant de choisir quelles règles appliquer, etc.
- D'autre part, le système conserve à tout moment dans sa mémoire de travail, une *base des faits* connus (ou d'instances de l'ontologie). Grâce à elle, le moteur d'inférence peut choisir les éléments de sa base de connaissances; par exemple, les règles à utiliser en fonction des faits disponible, de manière efficace et adaptée au problème posé. La base de faits s'enrichit au fur et à mesure que le moteur d'inférence déduit de nouveaux faits en appliquant des connaissances aux faits déjà connus.
- Le *moteur d'inférence* est le programme qui construit les raisonnements en puisant ses matériaux dans la base de connaissances et la base de faits. En examinant la base de faits, il détecte les connaissances intéressantes pouvant s'appliquer à certains faits, les enchaîne et construit un plan de résolution. Il déduit de nouveaux faits de ceux fournis au départ ou en cours d'interaction par l'usager. Indépendant du domaine, le moteur d'inférence rassemble les mécanismes de raisonnement qui vont exploiter la base de connaissances.

- À ces trois composantes essentielles s'ajoutent des *modules d'interface* eux aussi indépendants du domaine de connaissance. Grâce à eux, l'expert peut accéder facilement à la base de connaissances, la modifier en rectifiant une information inutile ou erronée, ou en ajoutant une précision. L'utilisateur peut quant à lui suivre le raisonnement du système dans un langage qui lui est naturel, poser des questions, demander des explications, sans avoir besoin d'acquérir une connaissance approfondie des systèmes experts ou de l'informatique.

Le domaine des systèmes à base de connaissances est, comme on peut le constater, un *domaine carrefour* qui emprunte des méthodes à la plupart des autres domaines de l'intelligence artificielle : représentation des connaissances, méthodes d'inférence et de résolution de problèmes, traitement du langage naturel, de la parole et des images dans les interfaces avec l'usager, méthodes d'induction et d'apprentissage pour l'acquisition des connaissances à partir d'experts ou de documents.

3.3 Modes de développement

On peut faire un parallèle entre, d'une part, l'utilisation et la construction d'un système à base de connaissances à l'aide d'un système générateur et, d'autre part, l'utilisation et la construction d'une base de données à l'aide d'un système de gestion de bases de données.

La réunion du moteur d'inférence et des modules d'interface qui lui sont associés forme ce que l'on appelle un *système générateur* ou *coquille de système expert* qui joue un rôle analogue à un logiciel de gestion de base de données comme ORACLE ou dBase, avec cette différence qu'il gère des connaissances plutôt que des données. Il suffit de lui ajouter une base de connaissances spécifique pour le rendre compétent dans un domaine quelconque. Quant aux faits ou données, ils sont fournis par l'utilisateur en cours d'interaction ou puisés dans une base de données.

Le système générateur est donc une sorte de progiciel grâce auquel en relativement peu de temps, sans recourir à un langage de programmation, on peut construire un système à base de connaissances appliqué à un domaine. Les compétences demandées sont alors une bonne capacité d'analyse des connaissances du domaine plutôt qu'une grande expertise informatique. La construction d'un système à base de connaissances à l'aide d'un système générateur repose sur l'approfondissement des diverses connaissances à l'œuvre dans un domaine et de leurs interrelations.

Il existe aujourd'hui des dizaines de systèmes essentiels qui facilitent la production de systèmes à base de connaissances et de systèmes experts. Plusieurs sont disponibles sur micro-ordinateurs. Ils se distinguent principalement par les modes de représentation des connaissances utilisés, les méthodes d'inférence et les interfaces qu'ils offrent à l'usager. La représentation des connaissances sous forme d'objet et les interfaces graphiques de type hypermédia se sont imposés. De plus, la plupart des systèmes de développement des SBC comportent des modules permettant d'importer des données d'un tableur ou d'une base de données ou d'afficher les résultats de la consultation sous forme textuelle et graphique. Certains fournissent des explications détaillées de leurs raisonnements et d'autres facilitent l'acquisition des connaissances.

3.4 Domaines d'application

Les systèmes à base de connaissances se répartissent pour la plupart entre quelques grandes catégories :

- Les *systèmes d'interprétation* permettent de construire une représentation symbolique : analyse de style ou de contenu, fabrication de résumés, etc. On y retrouve des systèmes de reconnaissance de la parole, d'analyse d'images et de textes et d'interprétation de signaux.
- Les *systèmes de prédiction* déduisent des conséquences probables à partir de situations données; par exemple, en météorologie, en démographie, en estimation des récoltes et en prévisions militaires.
- Les *systèmes de diagnostic* identifient les problèmes de fonctionnement d'un système à partir de données observables. Cette catégorie inclut, entre autres, les systèmes de diagnostic médicaux, d'appareils électroniques, de systèmes mécaniques ou logiciels.
- Les *systèmes de conception* mettent au point des configurations d'objets qui répondent aux contraintes du problème de conception; par exemple, des circuits électroniques ou des plans d'immeubles.
- Les *systèmes de planification* servent à mettre au point des séquences d'actions; par exemple, en construction automatiques de programmes informatiques, en robotique, en planification de projets ou en acheminement des marchandises.
- Les *systèmes de surveillance et de contrôle* observent et régissent de façon adaptative le comportement général d'un système. Pour ce faire, le système doit constamment interpréter la situation en cours, prédire la prochaine étape, identifier des problèmes et surveiller le fonctionnement du système. On trouve des applications au trafic aérien, à la gestion des affaires et au contrôle des missions.
- Les *systèmes de réparation* proposent des solutions à des pannes. Ils se basent sur leurs capacités de planification, de conception et de pronostic pour créer des spécifications ou des recommandations visant à corriger le problème diagnostiqué.
- Les *systèmes de formation* analysent les comportements des étudiants pour diagnostiquer des problèmes et y apporter des solutions. Généralement, ces systèmes construisent d'abord une description hypothétique des connaissances de l'étudiant servant à interpréter son comportement. Par la suite, ils identifient des faiblesses dans les connaissances de l'étudiant et en informent ce dernier ou le professeur. Souvent, ils préparent une interaction tutoriel visant à fournir l'information corrective à l'étudiant.

Actuellement, on peut affirmer que les « bons » systèmes à base de connaissances présentent en général les caractéristiques suivantes :

- ils possèdent des règles d'expertise qui évitent les recherches à l'aveuglette;
- ils peuvent saisir les principes fondamentaux d'un domaine spécialisé;
- ils peuvent fournir des explications de leurs raisonnements;
- ils peuvent traiter des problèmes difficiles relevant de domaines complexes.

Jusqu'à la fin des années 90, les systèmes experts ont souvent plusieurs limitations :

- un domaine d'expertise limité;
- une capacité limitée d'expression des faits et des relations;
- une connaissance limitée des problèmes et des méthodes de solution;
- des interfaces usagers peu développées;
- des explications sommaires et uniformes des étapes de raisonnement;
- une connaissance réduite de leur portée et de leurs limites;
- peu d'aide disponible au niveau de l'acquisition des connaissances;
- difficulté à combiner plusieurs sources de connaissances avec cohérence.

Depuis l'an 2000, les nouveaux développements sur le Web sémantique et le Web de données liées visent à corriger ces lacunes. Les chercheurs œuvrant dans ce domaine prévoient que la retombée la plus importante des systèmes experts est déjà la *codification des connaissances.* La possibilité qu'offre la technologie des systèmes experts d'obtenir une rétroaction et une évaluation des connaissances n'est pas moins importante. On arrive de cette façon à stimuler l'élaboration de nouvelles connaissances qui pourront réaliser certaines tâches avec plus de compétence que par le passé.

Conclusion

Ce texte d'introduction nous a permis de situer notre sujet, les systèmes à base de connaissances, dans le contexte plus large de l'intelligence artificielle, elle-même constituant une partie particulièrement innovatrice de la science informatique.

Déjà, le lecteur a pu constater l'importance des phénomènes en cause, non seulement sur le plan technologique, mais aussi sur ceux de l'emploi, de la vie dans les organisations et de la vie dans l'ensemble de nos sociétés.

Tout au long de ce texte, les deux concepts clés de représentation des connaissances et d'inférence ont été soulignés. Il nous reste à approfondir divers modes de représentation de ces connaissances ainsi que les méthodes nécessaires pour les acquérir, les modéliser, les transformer et les communiquer.

À retenir

1. L'intelligence artificielle est l'étude, par des moyens informatique, des processus qui composent l'intelligence humaine dans le but, d'une part, de mieux la comprendre et, d'autre part, de construire des logiciels plus « intelligents ».
2. Les facultés intellectuelles peuvent être modélisées par des systèmes de symboles qui représentent à la fois les connaissances et leurs méthodes de traitement.
3. Les méthodes d'intelligence artificielle se distinguent de l'informatique classique en ce qu'elles imitent les raisonnements humains, notamment leurs aspects heuristiques, et qu'elles utilisent des méthodes de déduction plutôt que de calcul.
4. Les ordinateurs modernes nous dépassent dans certains types de tâches, cependant ils ont beaucoup plus de difficultés que nous à effectuer d'autres tâches.
5. La représentation des connaissances est le mécanisme clé par lequel un système humain ou machine construit une représentation interne d'énoncés à propos du monde réel. Ces connaissances servent ensuite à faire des inférences, c'est-à-dire déduire de nouvelles connaissances à partir de faits connus.
6. Un système à base de connaissances est un programme informatique qui contient une représentation des connaissances d'un domaine, une ou des méthodes d'inférence et des modules de dialogue avec l'usager.
7. Un système expert est un logiciel capable, à partir de ses connaissances spécialisées, de résoudre des problèmes et de fournir des recommandations sur des questions relevant de son domaine d'expertise.
8. Le traitement du langage naturel est un domaine de l'intelligence artificielle qui permet l'interrogation en français de bases de données, l'analyse de textes, la génération de résumés et la traduction automatique d'une langue à une autre.
9. La reconnaissance des formes est un autre domaine de l'intelligence artificielle qui permet notamment la commande orale d'un ordinateur, l'analyse et la génération d'image et la reconnaissance de textes manuscrits.
10. L'apprentissage est un autre domaine de l'intelligence artificielle où on développe des programmes capables de modifier leurs comportements, notamment par l'induction de nouvelles connaissances générales à partir de faits particuliers.
11. La robotique est un domaine carrefour où on utilise les résultats des autres domaines de l'intelligence artificielle pour remplacer les programmes mécaniques de contrôle des machines par des programmes plus intelligents.