

# Histoire de l'Intelligence Artificielle

## 6- l'âge de raison de l'IA

Frédéric Fürst - [www.u-picardie.fr/~furst](http://www.u-picardie.fr/~furst)

Durant l'hiver connu par l'IA dans les années 70, la branche qui reste la plus active concerne les systèmes de raisonnement symbolique qui vont accoucher d'une réalisation exploitable industriellement : **les systèmes experts**.

Le début des années 80 voit également le redémarrage de travaux qui s'étaient englués dans les difficultés des années 60 : les **réseaux de neurones**, le **traitement automatique des langues**.

L'IA va cependant connaître un nouvel hiver à la fin des années 80, avant de rebondir dans les années 90, lorsque la banalisation de l'informatique va ouvrir de nouvelles perspectives de recherche et d'applications.

# Histoire de l'Intelligence Artificielle

## 6- l'âge de raison de l'IA

### 6.1- les systèmes experts



Limiter l'explosion combinatoire des systèmes de production passe par la restriction des programmes à des domaines limités et recelant peu d'ambiguïtés.

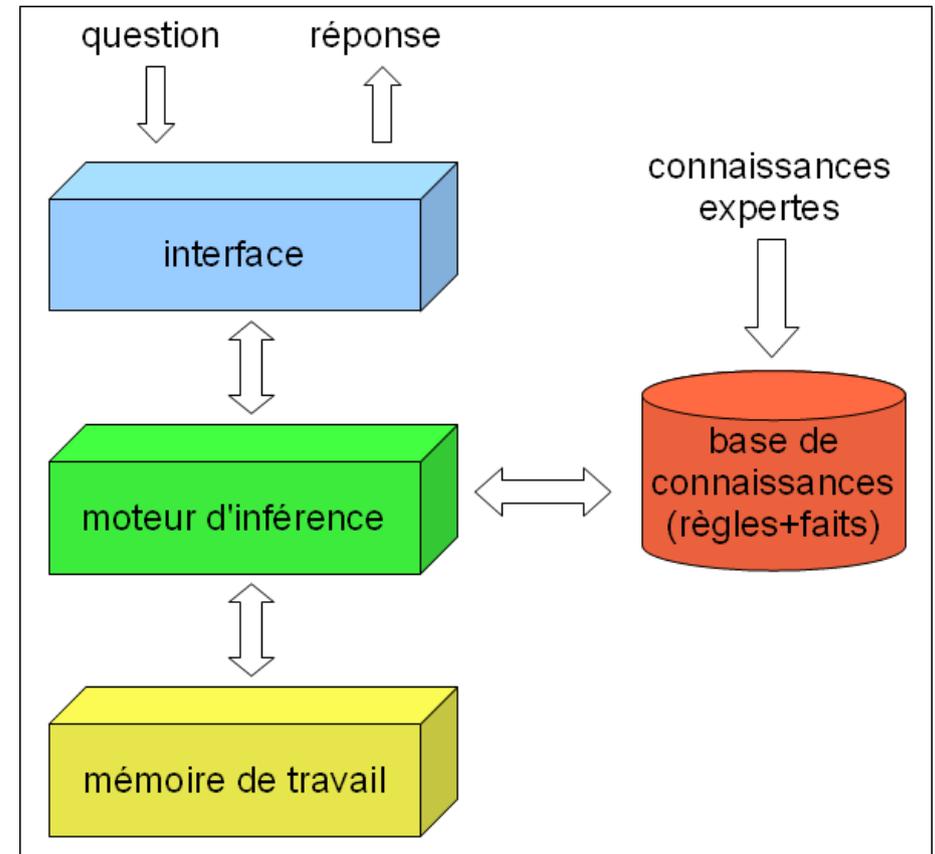
**Edward Feigenbaum** (né en 1936), ancien thésard de Simon, s'associe à Stanford avec Bruce Buchanan (informaticien), Joshua Lederberg (prix Nobel de médecine) et Carl Djerassi (chimiste inventeur de la pillule contraceptive) pour développer en 1965 le premier **Système Expert**.

L'idée de Feigenbaum est que l'intelligence est d'abord une affaire de connaissance, plus que de calcul ou de raisonnement. Il invente le terme **Knowledge Engineering** (Ingénierie des Connaissances) en 1977 et développe le Knowledge System Laboratory ([www-ksl.stanford.edu](http://www-ksl.stanford.edu)).

Le premier système expert, **DENDRAL** (Dendritic Algorithm), contient les connaissances des experts chimistes et les utilise pour répondre aux problèmes d'identification de composants chimiques à partir de mesures physiques (spectroscopie de masse, etc).

Un **système expert** encapsule de la connaissance sous forme de règles et de faits et dispose d'un mécanisme d'inférence lui permettant d'utiliser ces connaissances pour résoudre un problème.

Une règle est de la forme « *si tel fait est attesté alors effectuer telle action* ». Une action peut être l'ajout d'un fait, le retrait d'un fait ou la modification d'un fait existant dans la mémoire de travail.



Exemple de règle de DENDRAL :

**Si** le spectre de la molécule présente deux pics  $x_1$  et  $x_2$  tels que  $x_1 - x_2 = M + 28$  et  $x_1 - 28$  est un pic élevé et  $x_2 - 28$  est un pic élevé et au moins l'un des pics  $x_1$  et  $x_2$  est élevé  
**Alors** la molécule contient un groupe cétone

Des **moteurs d'inférences** efficaces vont rapidement être développés de façon indépendante. Ils sont alors réutilisables dans les différents systèmes experts.

Développer un système expert ne consiste pas à développer un mécanisme d'inférence, mais à **représenter les connaissances**. La première difficulté est de récupérer la connaissance auprès des experts humains, la deuxième de la représenter sous forme de règles.

Le problème de la *circularité* :

SI démarrage impossible ALORS absence d'essence  
SI absence d'essence ALORS démarrage impossible

Le problème de la *redondance* :

SI démarrage impossible ALORS absence d'essence  
SI démarrage impossible et batterie neuve ALORS absence d'essence

Le problème de l'*incompatibilité* :

SI démarrage impossible ALORS absence d'essence  
SI démarrage impossible et jauge d'essence > 0 ALORS présence d'essence



Un des systèmes experts les plus connus est **MYCIN**, développé à partir de 1972 par un thésard de Buchanan, **Edward H. Shortliffe**.

MYCIN est dédié au diagnostic de maladies infectieuses du sang (et des méningites dans une version ultérieure) et à la prescription médicale. C'est le premier à bien séparer le moteur d'inférence de la base de connaissances et à pouvoir **expliquer** ses raisonnements.

Le système initial comporte 200 règles mais atteint rapidement les 500 règles.

En 1978, un test en grandeur réelle est réalisé : MYCIN et 9 médecins, internes et universitaires réalisent des diagnostics et des prescriptions sur 80 patients atteints de méningites.

Les diagnostics et prescriptions sont ensuite évalués en aveugle par 8 spécialistes de la méningite et ... MYCIN obtient un score plus élevé que ceux des experts humains.

Les années 70 voient les **systèmes experts** se diversifier et sortir des laboratoires.

PROSPECTOR est développé en 1976 à Stanford pour l'**évaluation** de la richesse en minerais d'un site géologique.

XCON est développé en 1978 par Digital Equipment Corporation (DEC) pour la **configuration** d'ordinateurs **VAX-11** à partir de spécifications client.



En 1977, l'équipe de MacCarthy développe TAXMAN, un système expert capable de **conseiller** une entreprise sur la meilleure façon de payer le moins de taxes possible. Depuis, le système a été repris et étendu par diverses entreprises.

Au début des années 80, General Electric développe DELTA (Diesel Electric Locomotive Troubleshooting Aid), pour le **diagnostic** des pannes de locomotives.

Les **moteurs d'inférences** s'améliorent. En 1979, Charles R. Forgy invente l'algorithme RETE, qui accélère la détermination des règles applicables.

La série des moteurs OPS (Official Production System) aboutit au moteur CLIPS (C Language Integrated Production System) développé en 1984 par la NASA pour réaliser des systèmes experts de diagnostic sur les fusées et de pilotage de mission spatiale.

CLIPS est désormais open-source, comme de nombreux autres moteurs (*clipsrules.sourceforge.net*).



Des **fonctionnalités nouvelles** voient le jour, par exemple pour faciliter l'acquisition des connaissances. Le module TEIREISAS est ajouté dans MYCIN où, en cas de diagnostic erroné, il interagit avec le médecin pour apprendre de nouvelles règles ou/et modifier les règles existantes.

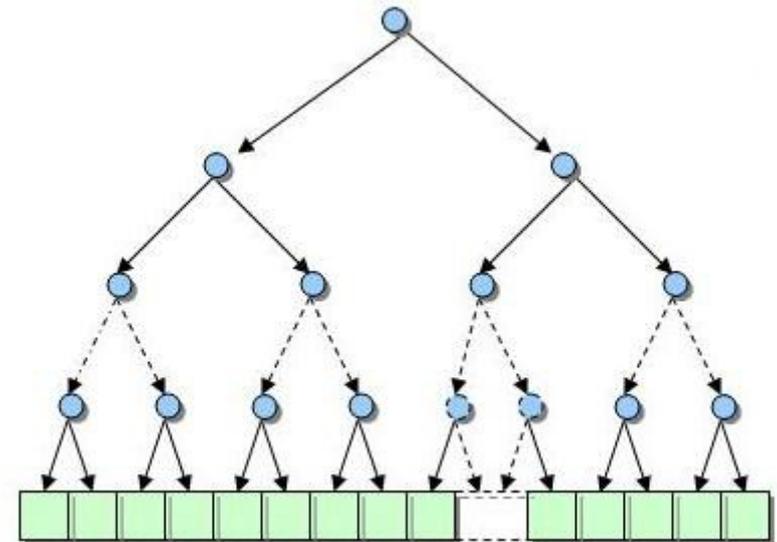
Les systèmes experts sont efficaces pour des tâches précises pour lesquelles aucune procédure algorithmique n'est connue, dans un domaine de connaissances réduit et technique.

Lancé en 1984, le projet **Cyc** vise à construire une base de connaissances portant sur toutes les connaissances de **sens commun**. La version 2.0 de l'OpenCyc contient des centaines de milliers de notions (concepts) et des millions d'assertions les concernant ([www.opencyc.org](http://www.opencyc.org)).



Le **jeu des 20 questions** consiste à trouver en maximum 20 questions un objet choisi par l'autre joueur. Un système expert implémentant le jeu sous forme d'un arbre de décision binaire peut discriminer en 20 questions  $2^{20}$  objets soit plus d'un million (voir par exemple le site [y.20q.net](http://y.20q.net)).

Tout le problème est d'organiser correctement les connaissances dans le système.



# Histoire de l'Intelligence Artificielle

## 6- l'âge de raison de l'IA

6.2- de nouveaux modèles pour le raisonnement  
et les connaissances

Les **logiques non classiques** sont prises en compte et permettent d'augmenter l'expressivité de la logique de Frege.

La **logique multivaluée** proposée par **Jan Lukasiewicz** (1878-1956) en 1920 prend ses valeurs de vérités sur l'intervalle réel  $[0,1]$  et redéfinit les opérateurs :

$$\neg A = 1 - A$$

$$A \wedge B = \min(A,B)$$

$$A \vee B = \max(A,B)$$

$$A \rightarrow B = \min(1, 1-A+B)$$



Des logiques multivaluées avec un nombre limité de valeurs de vérité ont été aussi proposées : la **logique trivalente** par exemple (vrai, faux, peut-être).



**Lotfi Zadeh** (né en 1921) a proposé en 1965 la **logique floue**, qui n'est pas une vraie logique mathématiquement parlant, mais utilise comme valeurs de vérité des sous-ensembles de  $[0,1]$ .

En logique classique, l'énoncé  $\neg A \rightarrow (A \rightarrow B)$  est toujours vrai, ce qui ne correspond pas à l'intuition. Clarence Irving Lewis (1883-1964), logicien, propose en 1912 d'utiliser un opérateur de **possibilité**, noté  $\diamond$ , et introduit l'implication stricte  $>$  :

$$A > B = \neg \diamond (A \wedge \neg B)$$

L'opérateur de **nécessité**, est défini par :  $A = \neg \diamond (\neg A)$ .

Ces opérateurs caractérisent les **logiques modales**. Rien n'interdit d'ajouter d'autres opérateurs, par exemple les opérateurs F (dans le futur) et P (dans le passé) utilisés dans les **logiques temporelles**.

Changer les axiomes de base de la logique classique permet également de développer d'autres logiques, comme la **logique intuitionniste**, qui n'accepte pas le principe du tiers-exclu,  $\vdash (A \vee \neg A)$ , ni le principe de démonstration par l'absurde,  $(S \wedge \neg A) \rightarrow \perp \vdash (S \rightarrow A)$ .

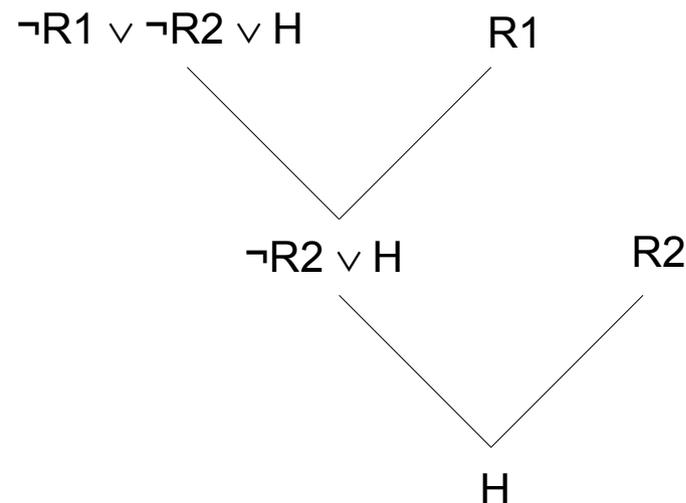


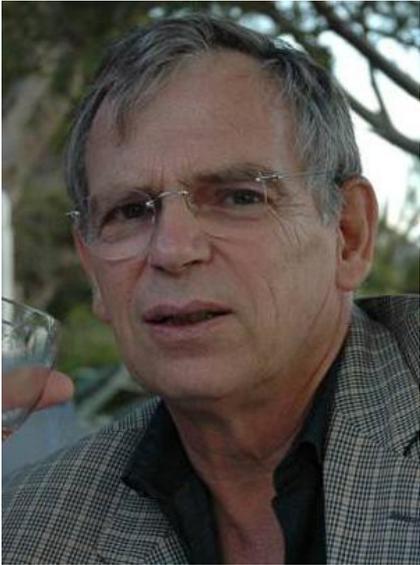
Pour faire face à l'explosion combinatoire dans les raisonnements, la **logique classique** est restreinte dans ses expressions et ses règles de raisonnement.

John Alan Robinson (né en 1930) propose en 1965 de restreindre les expressions logiques aux **clauses de Horn** qui sont de la forme  $R1 \wedge R2 \wedge R3 \dots \rightarrow H$ , ce qui peut également s'écrire  $(\neg R1 \vee \neg R2 \vee \neg R3 \dots) \vee H$

Pour raisonner efficacement à partir de telles clauses, il propose d'utiliser le **principe de résolution** :  $A \vee B, \neg A \vee C \vdash B \vee C$

Comment déduire H à partir de la règle  $\neg R1 \vee \neg R2 \vee H$  et des faits R1 et R2 ?





**Alain Colmerauer** (né en 1941) et Philippe Roussel (né en 1945) développent à Marseille en 1972 le langage Prolog (**Programmation Logique**), au départ pour traiter le langage.

Un programme Prolog est une suite de clauses de Horn sur lesquelles opère un mécanisme de raisonnement utilisant le principe de résolution.

Comme LISP, Prolog utilise massivement la structure de liste et est naturellement récursif.

```
factorielle(1,1) .  
factorielle(N,F) :- N=\=1,X is N-1,factorielle(X,Y),F is Y*N.
```

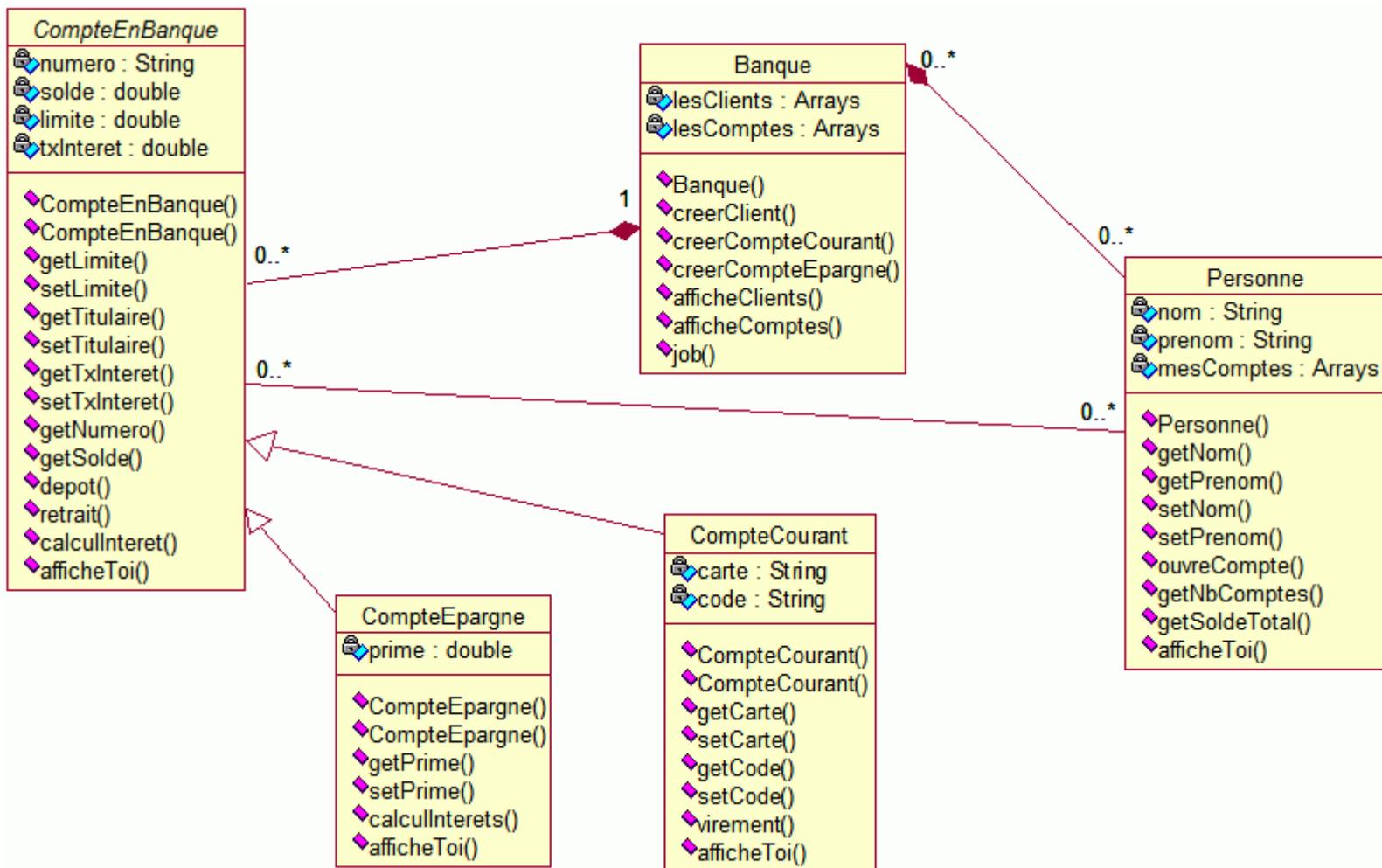
```
mylength([],0) .  
mylength([_|Y],L) :- mylength(Y,L2),L is L2 + 1.
```

Prolog devient un des langages standards de l'IA, avec LISP.

Les scruffies continuent à inventer des modèles de connaissances en marge de la logique.

En 1974 Minsky invente le modèle des **frames**. Une frame associe une description d'une situation sous forme d'attributs et des informations sur la façon dont la frame peut être utilisée (voir l'article *A Framework for Representing Knowledge*).

Ce modèle donnera naissance dans les années 1980 à la **programmation objet**.





Le programme **MARGIE** (Meaning Analysis, Response Generation and Inference on English) est développé en 1975 pour expérimenter son idée mais le typage des situations requiert la représentation du contexte.

En 1975, devenu professeur à Yale, Schank propose avec un psychologue de cette université, Robert Abelson, le modèle des **scripts**, qui représentent les situations au travers de séquences d'actions (*Scripts, plans and knowledge*. R.Schank and R. Abelson. 1975).

### Exemple du restaurant :

John went to a restaurant.  
He ordered a hamburger.  
It was cold when the waitress brought it .  
He left her a very small tip .



### *Scene 1: Entering*

PTRANS self into restaurant  
ATTEND eyes to where empty tables are  
MBUILD where to sit,  
PTRANS self to table  
MOVE sit down

### *Scene 2: Ordering*

PTRANS receive menu  
MTRANS read menu  
MBUILD decide what self want  
MTRANS order to waitress

En 1978, le programme **SAM** (Script Applier Mechanism), écrit par Richard Cullingford utilise des scripts d'accident de voiture pour comprendre le contenu d'article de presse sur le sujet.

### *Scene 3: Eating*

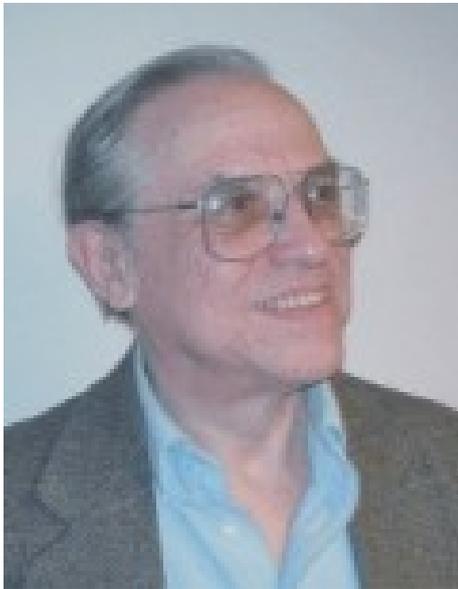
ATRANS receive food  
INGEST food

### *Scene 4: Exiting*

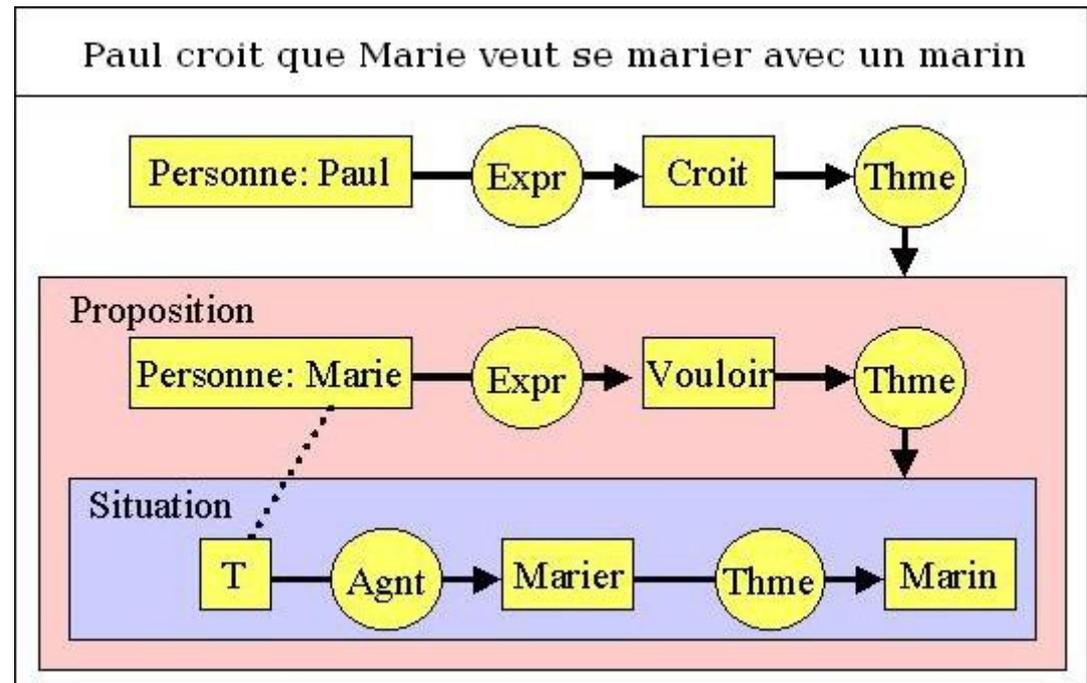
MTRANS ask for check  
ATRANS receive check  
ATRANS tip to waitress P  
TRANS self to cashier  
ATRANS money to cashier  
PTRANS self out of restaurant



Certains travaux mêlent la rigueur mathématique de la logique et des modèles plus exotiques.



En 1984, **John Sowa**, chercheur à IBM, propose un modèle de représentation de connaissances de type réseau sémantique dont le raisonnement repose sur des mécanismes de la théorie des graphes : les **graphes conceptuels**.



Moins expressifs que la logique des prédicats, le point fort des graphes conceptuels est d'offrir un formalisme intuitif.

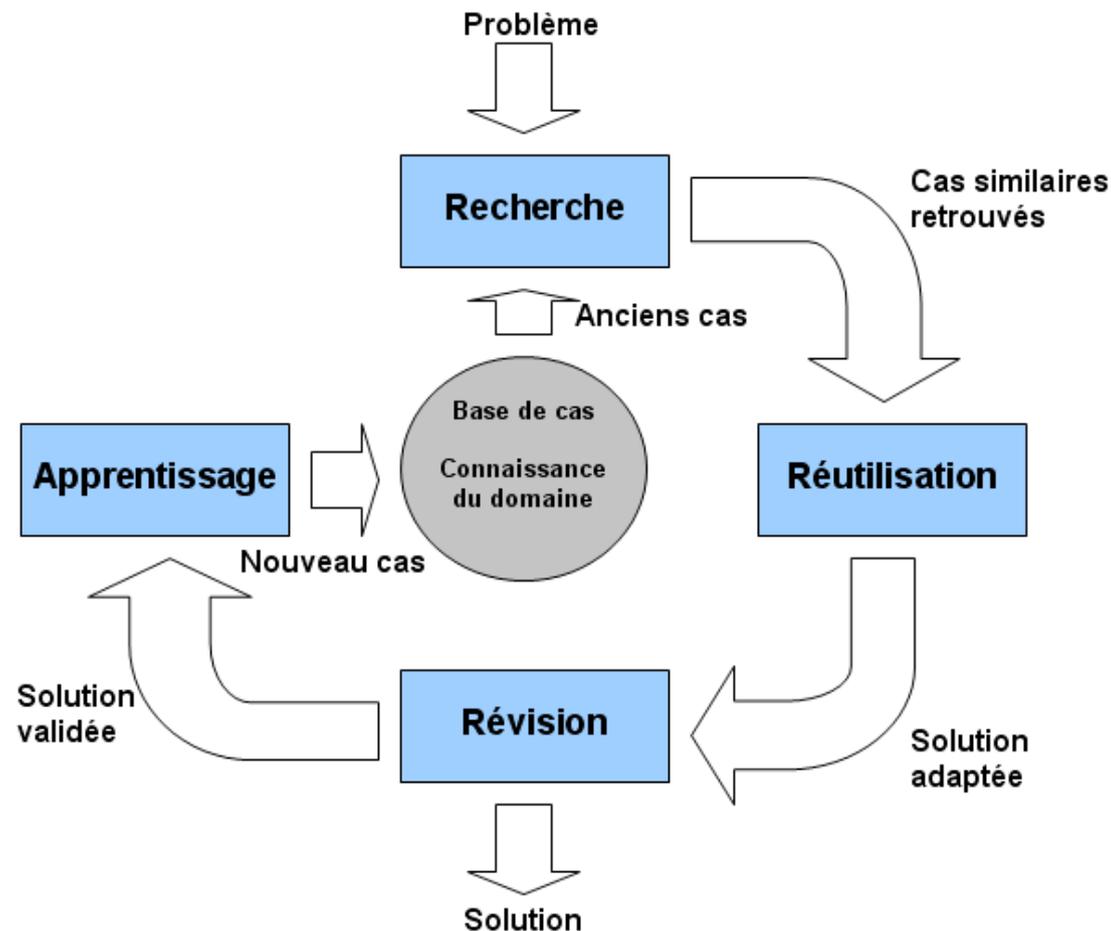
La combinaison des frames et des réseaux sémantiques va aboutir à une nouvelle famille de logiques : les **logiques de description**.

Un peu moins expressives que la logique des prédicats, leurs mécanismes de raisonnements sont plus efficaces.

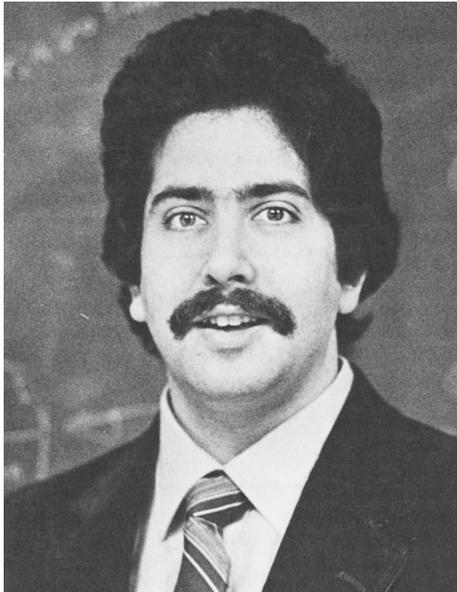
| Axiom                   | DL Syntax                         | Example  |
|-------------------------|-----------------------------------|--|
| subClassOf              | $C_1 \sqsubseteq C_2$             | Human $\sqsubseteq$ Animal $\sqcap$ Biped                      |
| sameClassAs             | $C_1 \equiv C_2$                  | Man $\equiv$ Human $\sqcap$ Male                               |
| subPropertyOf           | $P_1 \sqsubseteq P_2$             | hasDaughter $\sqsubseteq$ hasChild                             |
| samePropertyAs          | $P_1 \equiv P_2$                  | cost $\equiv$ price  |
| disjointWith            | $C_1 \sqsubseteq \neg C_2$        | Male $\sqsubseteq \neg$ Female                                 |
| sameIndividualAs        | $\{i_1\} \equiv \{i_2\}$          | {President_Bush} $\equiv$ {G_W_Bush}                           |
| differentIndividualFrom | $\{i_1\} \sqsubseteq \neg\{i_2\}$ | {john} $\sqsubseteq \neg$ {peter}                              |
| inverseOf               | $P_1 \equiv P_2^-$                | hasChild $\equiv$ hasParent <sup>-</sup>                       |
| transitiveProperty      | $P^+ \sqsubseteq P$               | ancestor <sup>+</sup> $\sqsubseteq$ ancestor                   |
| uniqueProperty          | $T \sqsubseteq \leq 1 P.T$        | T $\sqsubseteq \leq 1$ hasMother.T                             |
| unambiguousProperty     | $T \sqsubseteq \leq 1 P^-.T$      | T $\sqsubseteq \leq 1$ isMotherOf <sup>-</sup> .T              |
| range                   | $T \sqsubseteq \forall P.C$       | T $\sqsubseteq \forall$ hasParent.Human                        |
| domain                  | $T \sqsubseteq \forall P^-.C$     | T $\sqsubseteq \forall$ hasParent <sup>-</sup> .Human          |
| $i \text{ type } C$     | $i : C$                           | john : Man   |
| $i_1 P i_2$             | $\langle i_1, i_2 \rangle : P$    | $\langle \text{john}, \text{peter} \rangle : \text{hasParent}$ |

Parallèlement à l'évolution des modèles de représentation des connaissances, les **techniques de raisonnement** évoluent et se diversifient.

**Schank** propose au début des années 80 l'idée du **raisonnement à partir de cas** : le plus souvent, on ne raisonne pas à partir de règles générales, de connaissances que l'on a abstraites, mais directement à partir de cas et en utilisant des **analogies**.



L'**apprentissage symbolique** a pour principe de donner la capacité au système de gérer ses connaissances (réflexivité) et les moyens de les faire évoluer.



En 1983, **Doug Lenat** écrit le programme EURISKO capable d'inventer de nouvelles heuristiques en utilisant des **méta-heuristiques**.

Il l'utilise pour jouer à un jeu informatique de guerre interstellaire où les joueurs doivent concevoir leurs vaisseaux spatiaux. EURISKO gagne en concevant un vaisseau très original et bizarre mais plus adapté au jeu que ceux créés par les joueurs humains.

C'est une des premières tentatives pour doter les machines d'une réelle créativité. Mais ce système ne résout pas le problème de la régression infinie.

L'**apprentissage** qui vise à permettre aux programmes de s'améliorer automatiquement, va devenir une des branches importantes de l'IA à partir des années 80. En plus des réseaux de neurones, de nombreux modèles mathématiques sont utilisés et pour chaque modèle, de nombreux algorithmes d'apprentissage.

Les principes de l'apprentissage automatique s'inspirent des processus d'apprentissage humains :

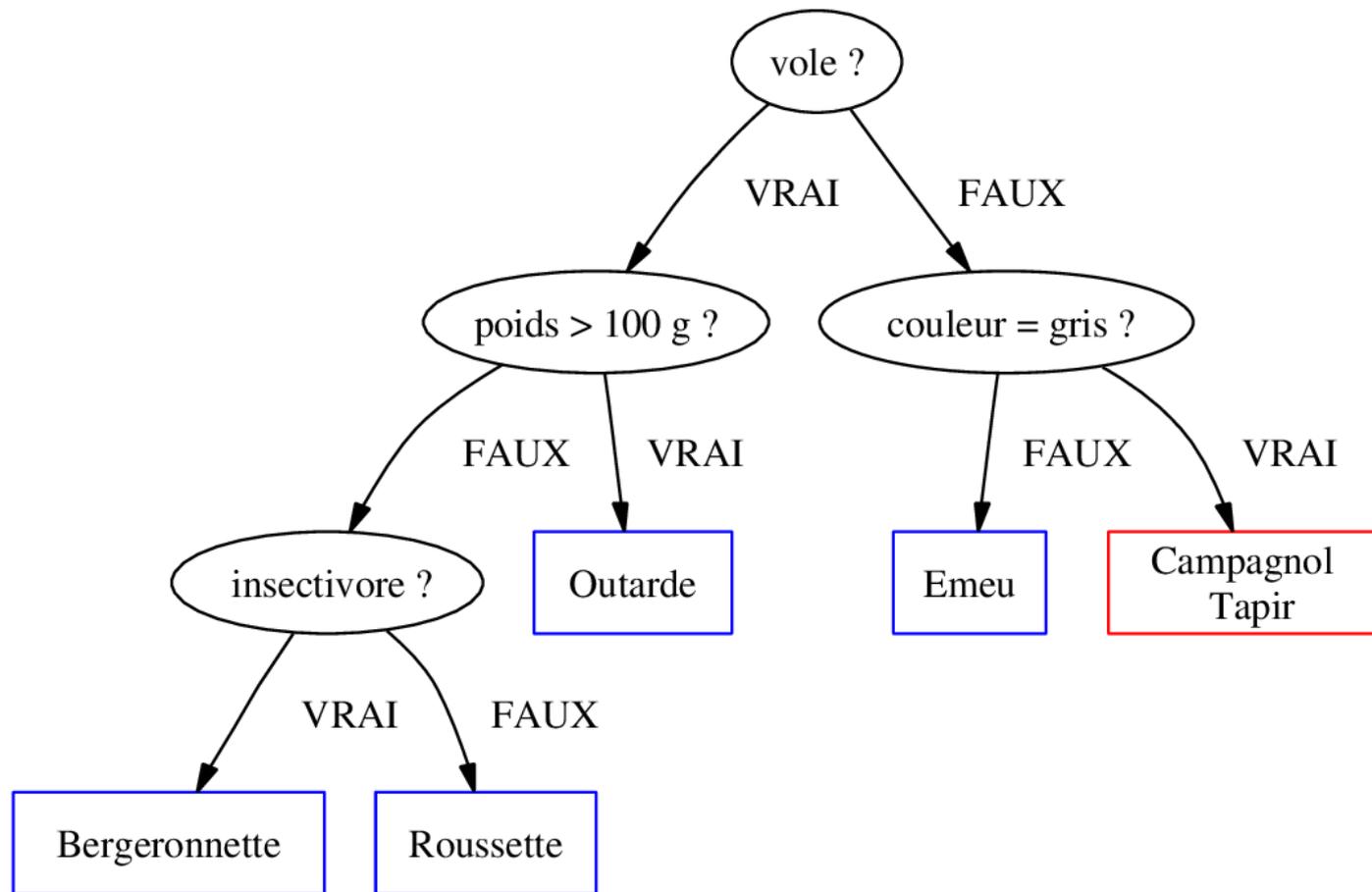
- **apprentissage supervisé** : le système dispose d'exemples sous forme de couples (entrée, sortie attendue), et parfois de contre exemples et est capable de construire statistiquement une règle de décision qu'il appliquera par analogie.
- **apprentissage non supervisé** : le système dispose d'entrées uniquement et les classe selon des critères, essentiellement statistiques, qu'il appliquera pour discriminer les nouvelles entrées.
- **apprentissage par renforcement** : le système dispose d'entrées et doit tester des sorties. Un évaluateur (informatique ou non) peut noter les sorties testées. Il s'agit de reproduire l'apprentissage par essai-erreur.



Les **arbre de décision** constituent un exemple simple de modèle utilisable pour l'apprentissage automatique. **Ross Quinlan** propose en 1983 le premier algorithme pour l'apprentissage des arbres de décision, nommé ID3.

*Exemple d'apprentissage supervisé : on veut développer un système capable d'apprendre à reconnaître des animaux et on dispose de descriptions d'animaux.*

| VOLE | POIDS | COULEUR       | NUTRITION   | PEAU   | ANIMAL        |
|------|-------|---------------|-------------|--------|---------------|
| oui  | 1kg   | roux          | granivore   | plumes | outarde       |
| oui  | 20g   | gris et jaune | insectivore | plumes | bergeronnette |
| non  | 100kg | noir et blanc | omnivore    | plumes | émeu          |
| non  | 5g    | gris          | granivore   | poils  | campagnol     |
| non  | 40kg  | gris          | herbivore   | poils  | tapir         |
| oui  | 60g   | noir          | frugivore   | poils  | roussette     |



Le problème est de déterminer comment **construire l'arbre de façon optimale** pour que son parcours soit efficace, c'est-à-dire comment choisir les critères de test et comment les ordonner de façon optimale.

D'autres modèles, souvent probabilistes, ont été développés (chaines de Markov, réseaux bayésiens, ...).

# Histoire de l'Intelligence Artificielle

## 6- l'âge de raison de l'IA

### 6.3- l'intelligence artificielle à l'école de la biologie



En 1982 les travaux du physicien **John Hopfield** relancent les recherches sur les **réseaux de neurones**.

Hopfield propose un modèle de réseau particulier (totalement connecté) et un nouveau principe d'apprentissage, la *rétropropagation du gradient*, qui permettent de résoudre le problème soulevé par Minsky et Papert. Mais d'autres problèmes encore plus ardues demeurent.

De nombreuses **applications industrielles** voient le jour : diagnostic (voitures Renault, photocopieurs Canon, ...), compression de données (JVC), prévision de consommation d'eau (Lyonnaise des Eaux, CGE), d'électricité (EDF), de trafic routier (Cofiroute) ou de cours boursiers, reconnaissances des formes, etc.

Par exemple, les **machines de tri du courrier** utilisent des réseaux de neurones formels pour reconnaître les chiffres des codes postaux sur les enveloppes.



Créer un cerveau artificiel à base de neurones formels reste cependant utopique.

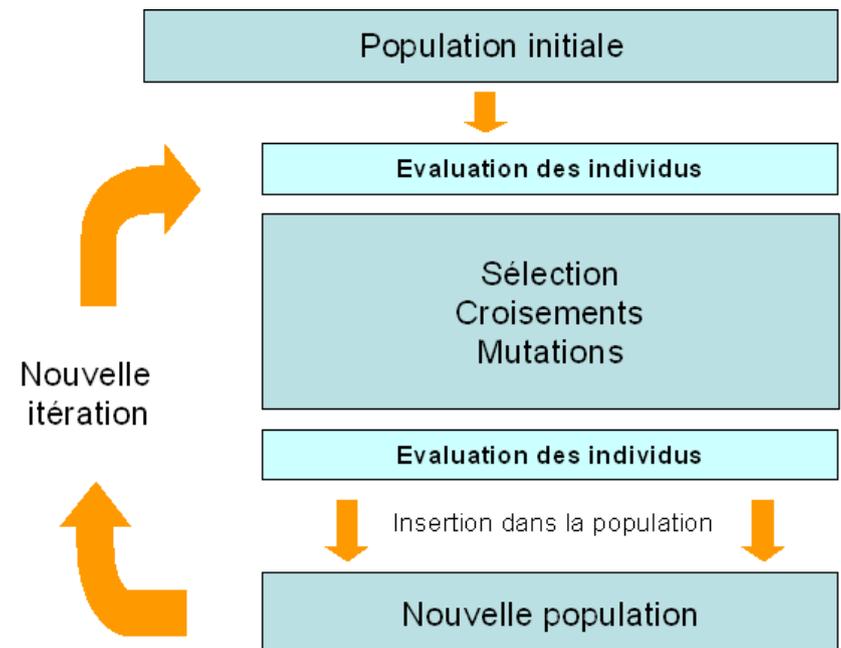


**Lawrence J. Fogel** (1928-2007) s'inspire des travaux de Charles Darwin sur l'évolution des espèces et propose en 1966 la **programmation évolutionnaire**.

Les **algorithmes génétiques** sont une classe particulière d'algorithmes évolutionnaires introduits en 1975 par John H. Holland.

Pour obtenir un programme réalisant une tâche complexe on part d'une population de programmes simples et on les fait évoluer jusqu'à ce qu'un programme réalisant la tâche considérée apparaisse dans la population.

Les algorithmes évolutionnaires sont utilisés pour développer des programmes de jeux performants (dames, jeux vidéos, etc) ou pour apprendre à un robot à exécuter des tâches qu'il ne sait pas réaliser (PathFinder, Aibo).



L'étude du comportement des **insectes sociaux** (fourmis en particulier) inspire également les chercheurs.

Les **systèmes multi-agents** sont basés sur le principe suivant : quand un problème est trop complexe pour être traité par un seul programme, la collaboration de plusieurs programmes peut permettre la résolution du problème.



Les systèmes multi-agents ont trouvé de nombreuses applications industrielles (simulation, contrôle de processus industriels complexes, surveillance de patients, fouille du Web, commerce électronique, etc). Mais de tels systèmes ne sont pas considérés comme intelligents.

# Histoire de l'Intelligence Artificielle

## 6- l'âge de raison de l'IA

### 6.4- la deuxième crise de l'IA

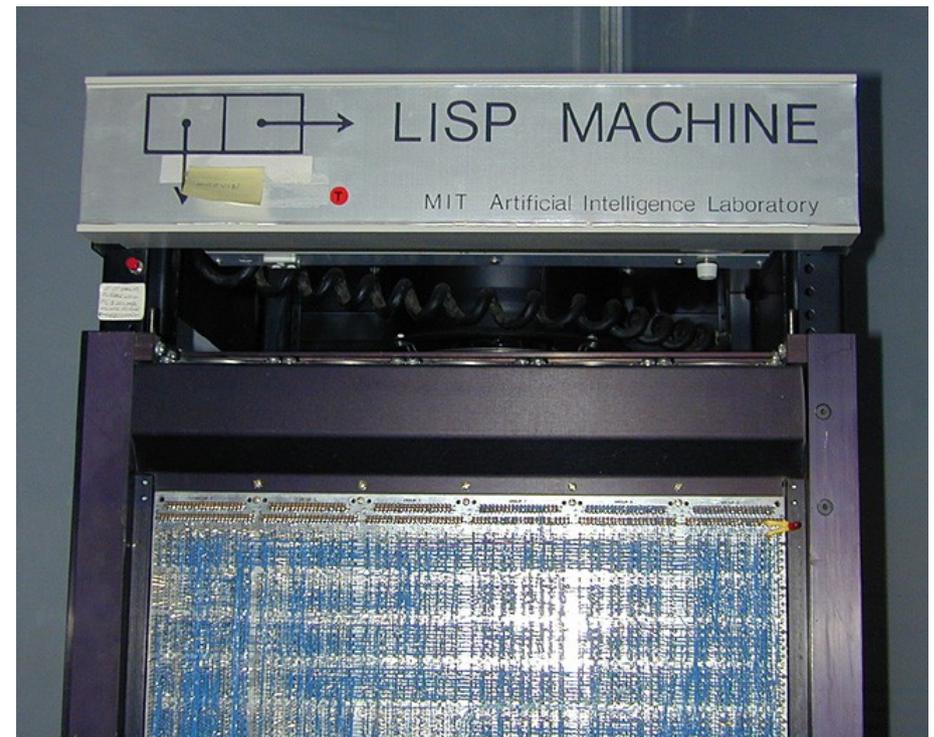
Les années 1980 voient un retour en grâce de l'IA, grâce en particulier au succès des systèmes experts et aux applications industrielles.

En 1981, le gouvernement japonais décide de lancer le Japon dans la course à l'IA. Le projet d'ordinateur de **cinquième génération** est lancé. Il bénéficiera en 10 ans de 500 millions de dollars et utilise préférentiellement le langage Prolog.

Aux USA, la DARPA (Defense Advanced Research Projects Agency) réplique avec le programme **Strategic Computer Initiative**, qui recevra un milliard de dollars.

Mais de nouvelles désillusions apparaissent à la fin des années 1980.

En 1987, les **machines LISP**, optimisées pour l'exécution de code LISP, sont abandonnées face à l'essor du Personal Computer (PC).



En 1988, face au manque de résultat, le financement du Strategic Computer Initiative est stoppé.

En 1990, le projet de cinquième génération est également arrêté. Il n'a produit que quelques avancées en informatique, **architectures parallèles** en particulier.



La reconnaissance de l'IA au niveau industriel est mitigée : IBM s'oppose depuis toujours à tout ce qui est étiqueté IA, alors que Xerox a mis un moment sur les machines LISP et que DEC a lancé quelques projets de systèmes experts.

# Histoire de l'Intelligence Artificielle

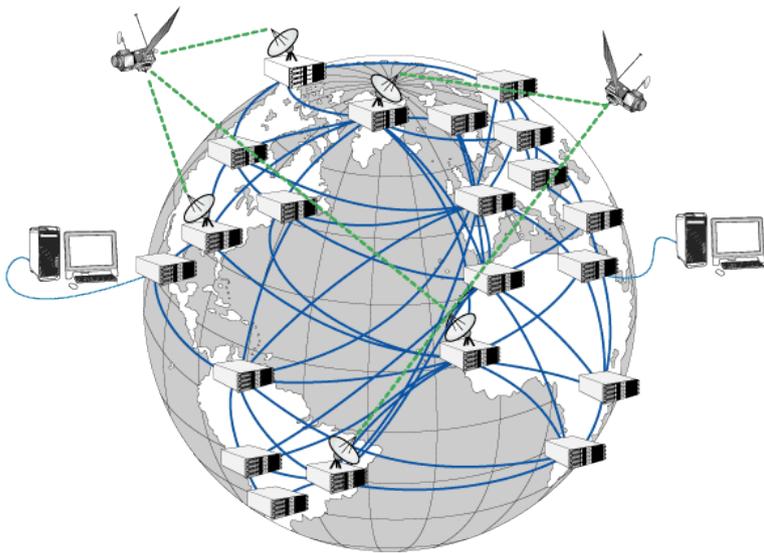
## 6- l'âge de raison de l'IA

### 6.5- l'IA à l'âge de l'informatique de masse

Les années 1990 voient l'informatique se diffuser massivement dans le monde professionnel mais également dans le **grand public**.



Les **ordinateurs personnels** deviennent courants, et sont de plus en plus puissants et compacts.



L'**Internet** fait des ordinateurs un des principaux outils de communication,

La **robotique** continue à se développer et fait même son entrée dans le grand public.



Certains ont pu voir dans le **Web** l'ébauche d'une intelligence semi-artificielle, mêlant machines et humains. Mais le Web est surtout un terrain d'application idéal pour l'IA.

La popularisation du Web rend nécessaire d'introduire de l'intelligence dans les applications, par exemple dans les **moteurs de recherche**.

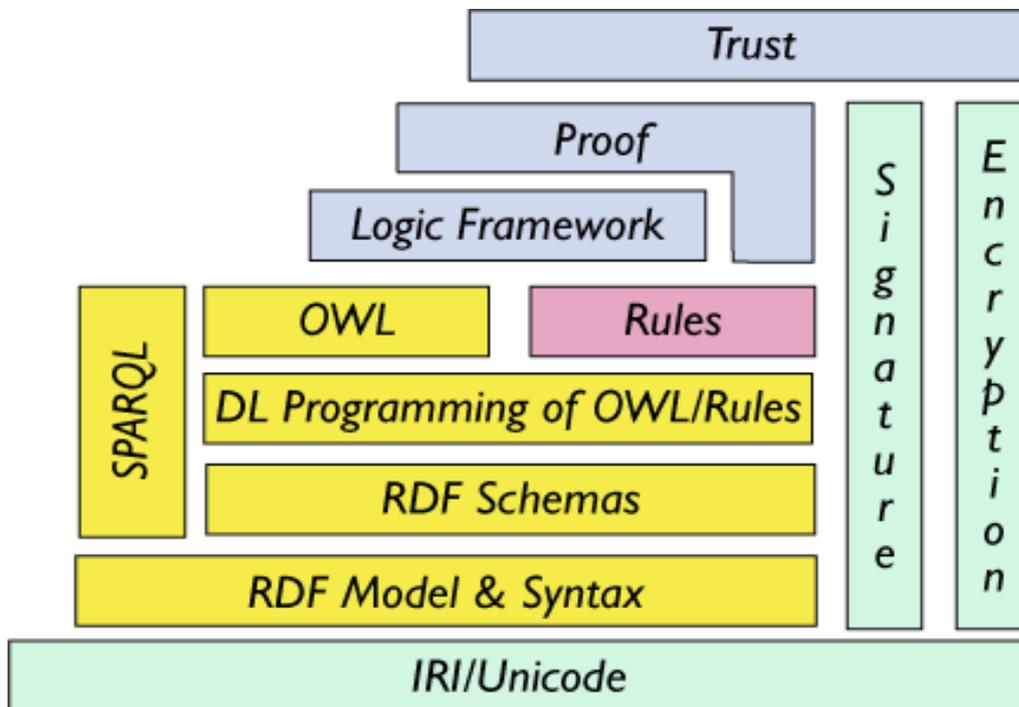
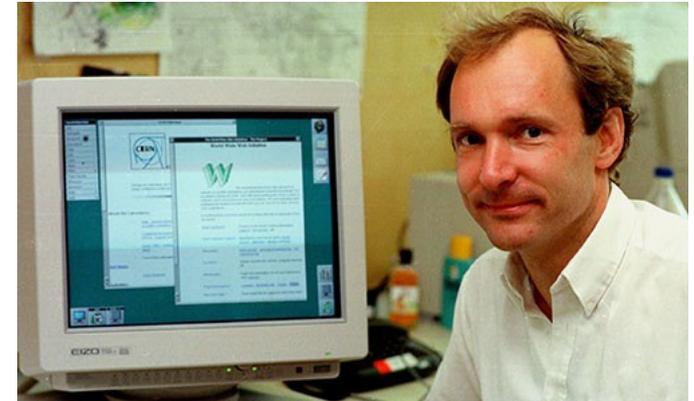
Google et consorts, le degré 0 de la recherche documentaire numérique :

- ▶ impossible d'expliquer vraiment ce qu'on cherche
- ▶ beaucoup de bruit dans les résultats car Google indexe tout
- ▶ les réponses sont triées de façon non pertinente (taux de référencement)

Idéalement, un moteur de recherche devrait avoir les mêmes capacités qu'un libraire ou un bibliothécaire humain sur les documents papier :

- ▶ un libraire est capable de comprendre des requêtes complexes
- ▶ un libraire n'a pas en tête tous les documents écrits depuis toujours, mais uniquement les livres qui peuvent présenter un intérêt
- ▶ un libraire ne propose pas uniquement les livres dont tout le monde parle, il connaît le contenu, peut poser des questions complémentaires, peut proposer des livres proches de ce que cherche le lecteur, etc

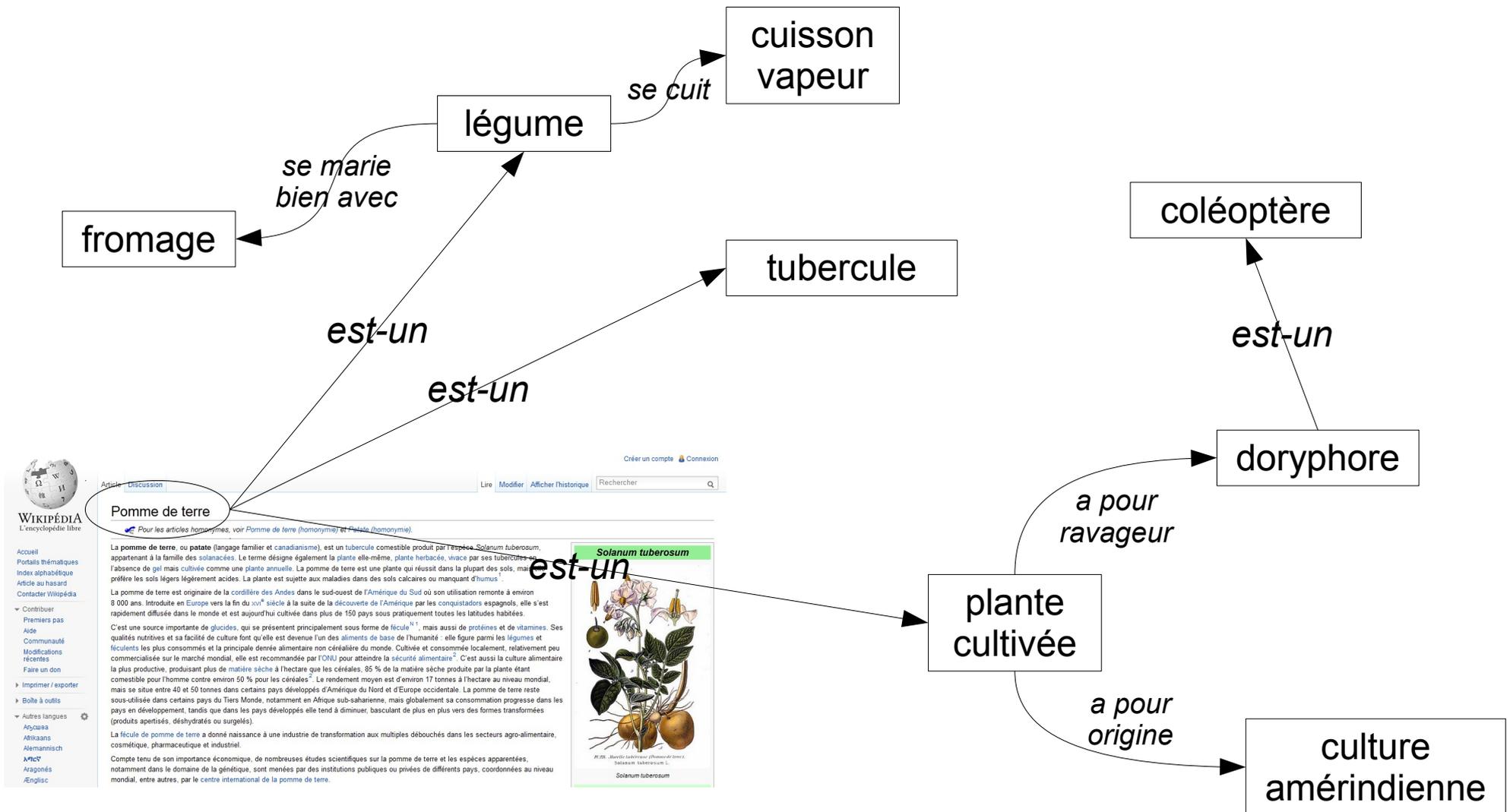
**Tim Berners-Lee**, qui a inventé le World Wide Web en 1989, lance au début des années 2000 le projet de **Web Sémantique** (ou Web 3.0).



L'objectif du Web Sémantique est d'ajouter au Web existant une couche de connaissances pour donner plus d'« intelligence » aux applications tournant sur le Web, en particulier aux moteurs de recherche.

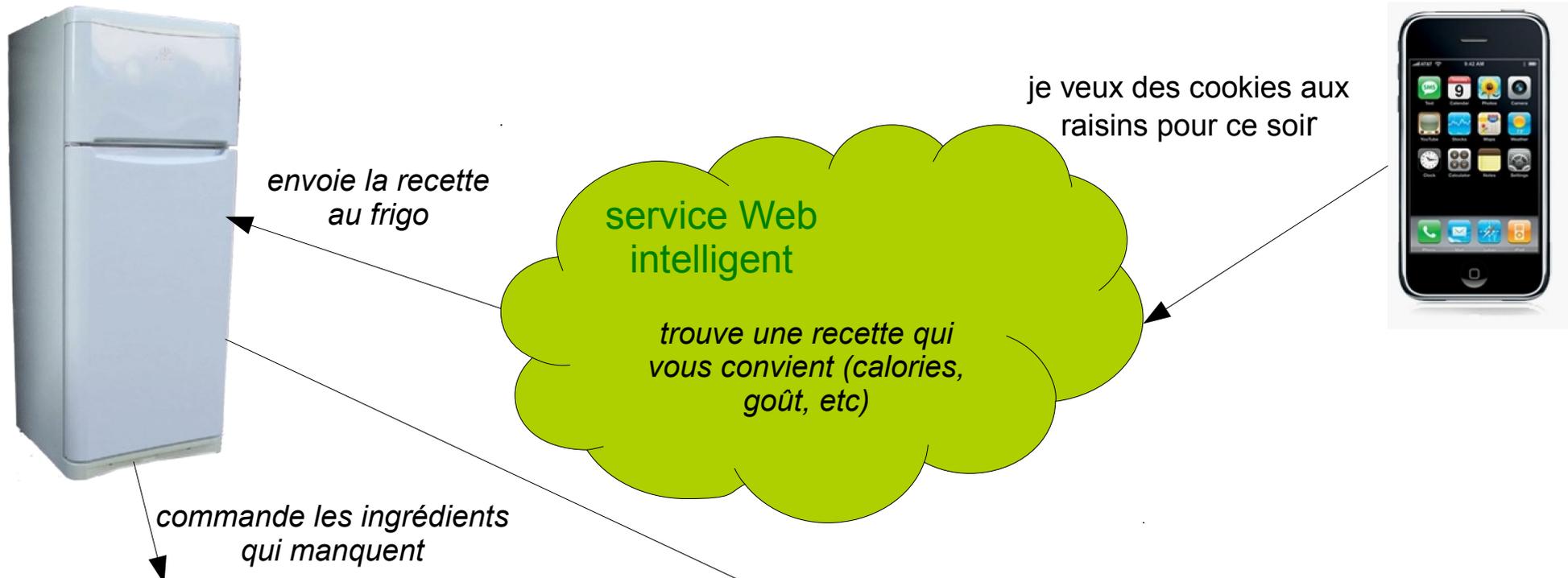
Le **Web des données** (Linked Data) consiste à ajouter au dessus des pages Web des représentations du sens du contenu des pages.

Cette sémantique des termes s'exprime par des **concepts** (au sens d'Aristote) et des **relations** entre ces entités.





Le **Web des objets** consiste à lier les objets électroniques au Web. L'intelligence ajoutée au Web à l'aide d'ontologies rend alors les objets « intelligents ».



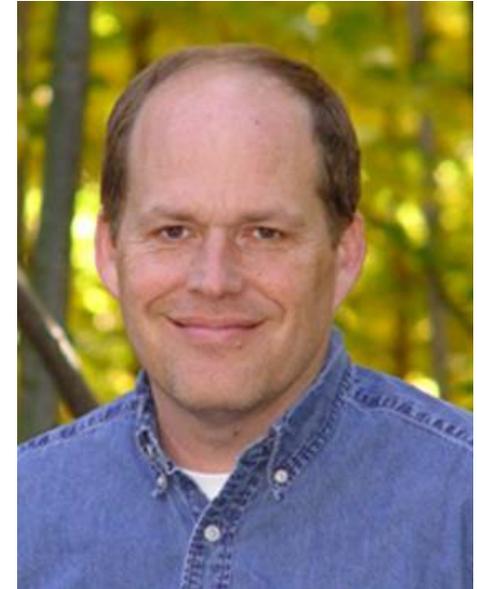
*quand les ingrédients sont livrés, ordonne au robot cuisinier de préparer les cookies*



Il ne s'agit plus de trouver LA solution unique au problème de l'IA, mais de combiner différentes techniques pour améliorer les performances.

Le projet **SOAR** (State, Operator And Result) est mené à partir de 1983 par **John Laird**, Alan Newell et Paul Rosenbloom.

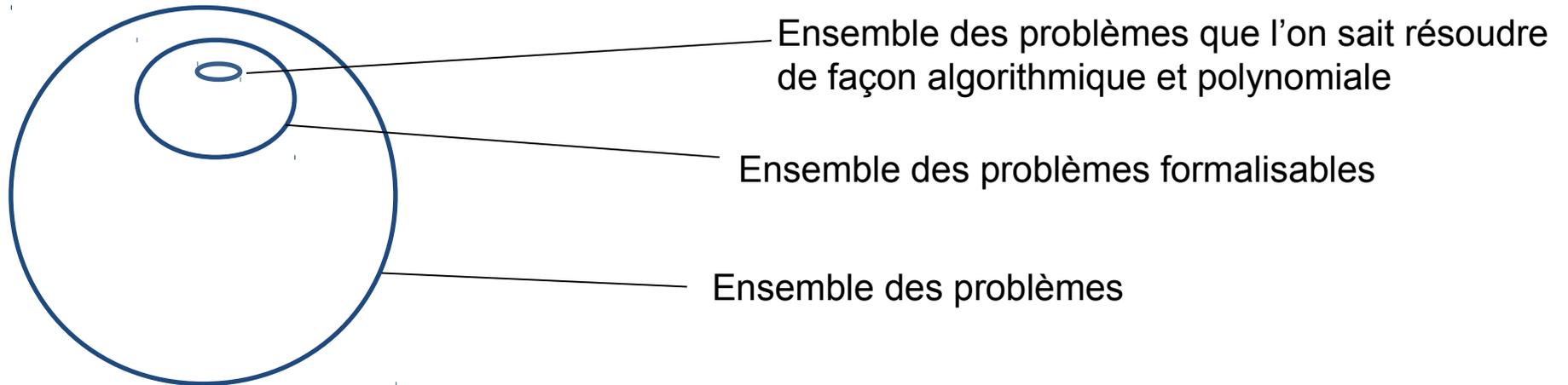
Le cœur de SOAR est un **système expert** de résolution de problème, auquel s'ajoutent des modules de **mémoire de travail**, d'**apprentissage** et de **résolution heuristique** qui sont utilisés par le système quand les connaissances disponibles dans le système expert ne suffisent pas.



Mais dans certains cas, recourir à des techniques simples est plus pertinent que de construire des modèles complexes.

Par exemple, en traduction automatique, l'**analyse statistique** des textes se révèle finalement plus efficace que l'analyse grammaticale (Google translator utilise l'analyse statistique).

L'objectif initial de l'IA, peut être trop ambitieux, laisse la place à une nouvelle approche : l'IA, que certains se mettent à désigner comme **Informatique Avancée**, s'attaque aux problèmes que l'approche algorithmique classique ne peut résoudre.



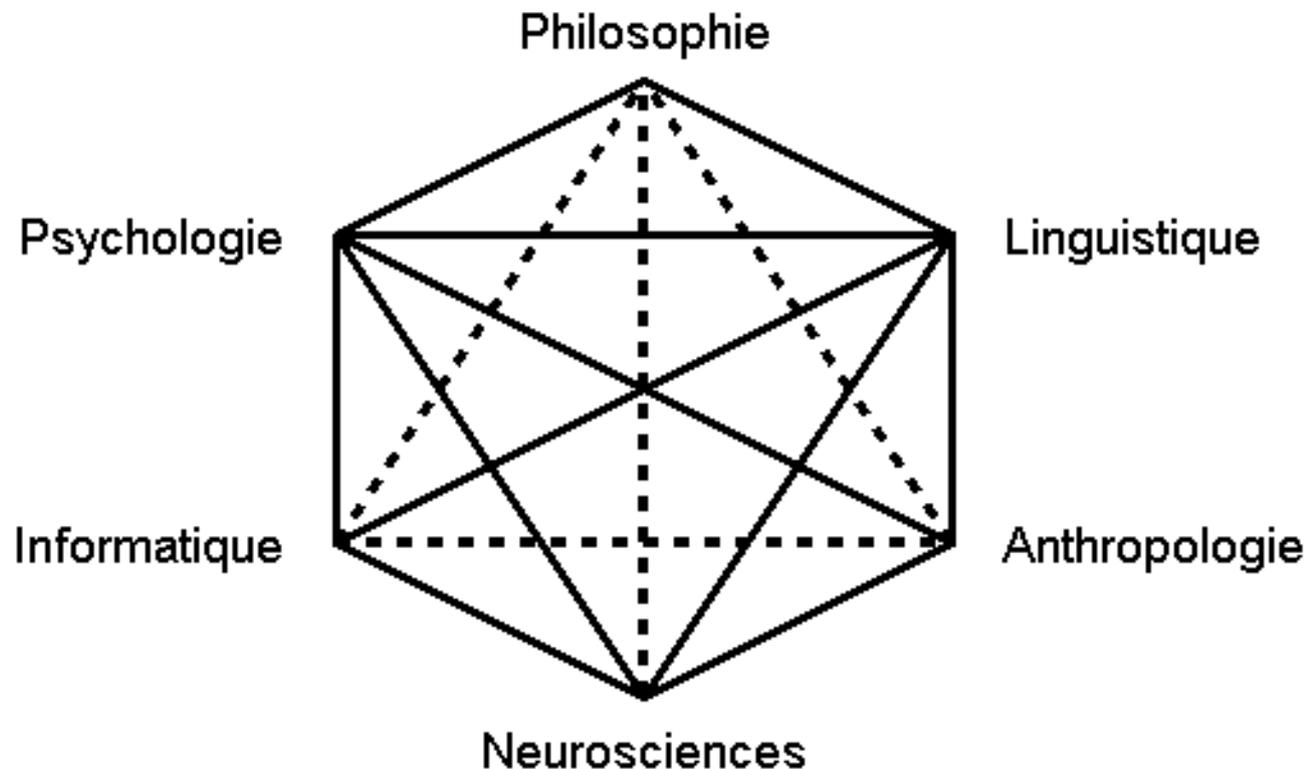
Problèmes formels résolus par **algorithmes polynomiaux** : résolution d'équations en variables entières, tester la connexité d'un graphe, ...

Problèmes formalisables résolus par **algorithmes non polynomiaux** : problème du voyageur de commerce, problème du sac à dos, diagnostics, ...

Problèmes peu formalisables résolus par **systèmes heuristiques**, réseaux de neurones, etc : gagner aux échecs, reconnaître un visage, ...

Mais l'objectif initial consistant à développer des machines aussi intelligentes que l'être humain continue à perdurer. Pour le résoudre, il est cependant nécessaire d'en savoir plus sur l'**intelligence humaine**.

L'IA s'inscrit de plus en plus dans le cadre des **sciences cognitives**, qui visent surtout à mieux comprendre l'intelligence humaine.



En 2009, le MIT a lancé le *Mind Machine Project* consistant à revenir 30 ans en arrière pour repenser les bases de l'IA.

Les déboires de l'IA n'ont pourtant pas entamé le **rêve initial** de 1956.



La **Robocup**, coupe du monde football pour robots, se tient tous les ans depuis 1997.

Elle a pour objectif d'arriver vers 2050 à ce qu'une équipe de robots batte une équipe humaine au football ([www.robocup.org](http://www.robocup.org)).

L'université de Carnegie-Mellon a lancé en 2010 le projet **Read the Web** où un programme lit indéfiniment les pages Web et en extrait des informations pour constituer une sorte de système expert couvrant toutes les connaissances humaines (<http://rtw.ml.cmu.edu/rtw/>).

Récemment a été lancée l'idée d'**autonomous computing** (logiciels autonomes, en particulier capables de gérer les pannes grâce aux connaissances qu'ils ont sur leur propre fonctionnement) et l'Union Européenne a lancé en 2010 un projet à long terme intitulé *Au delà des systèmes autonomes, le défi de la conscience* ([www.chistera.eu/call-2010](http://www.chistera.eu/call-2010)).

L'**astronautique** et l'**intelligence artificielle** ont suivi la même trajectoire : après des débuts fulgurants dans les années 1950/60, la conquête spatiale, comme l'IA, marque le pas depuis plusieurs décennies.

Mais l'épopée spatiale des années 1950/60 a changé l'idée que l'on se faisait de notre place dans l'univers. Ses retombées technologiques sont également nombreuses.

De même, l'IA a fait évoluer les idées sur notre intelligence et a apporté de nombreuses contributions théoriques et techniques à l'informatique.

Plus fondamentalement l'IA et l'astronautique sont le résultat de **rêves** et de fantasmes profondément ancrés en nous, ce qui devrait leur assurer un bel avenir.

