

# *IA et Nous*

## *Systemes experts*

# Les deux approches de l'IA

*Quand on veut modéliser un système, deux voies :*

- *Cognitive ou Symbolique* : modéliser le comportement
- *Connexionniste* : modéliser les mécanismes

# *L'approche cognitive ou symbolique*

*Repose sur le fait de pouvoir créer des programmes qui pensent comme les humains.*

*Comment pensent les êtres humains ?*

*Comment fonctionne l'esprit humain ?*

*Puis comment reproduire/simuler le raisonnement humain ?*

*Les **sciences cognitives** associent des **modèles informatiques** et les techniques expérimentales de la **psychologie**.*

# *Approches de l'IA*

**Connexionnisme**



***Neurosciences***

***Réseaux de neurones***

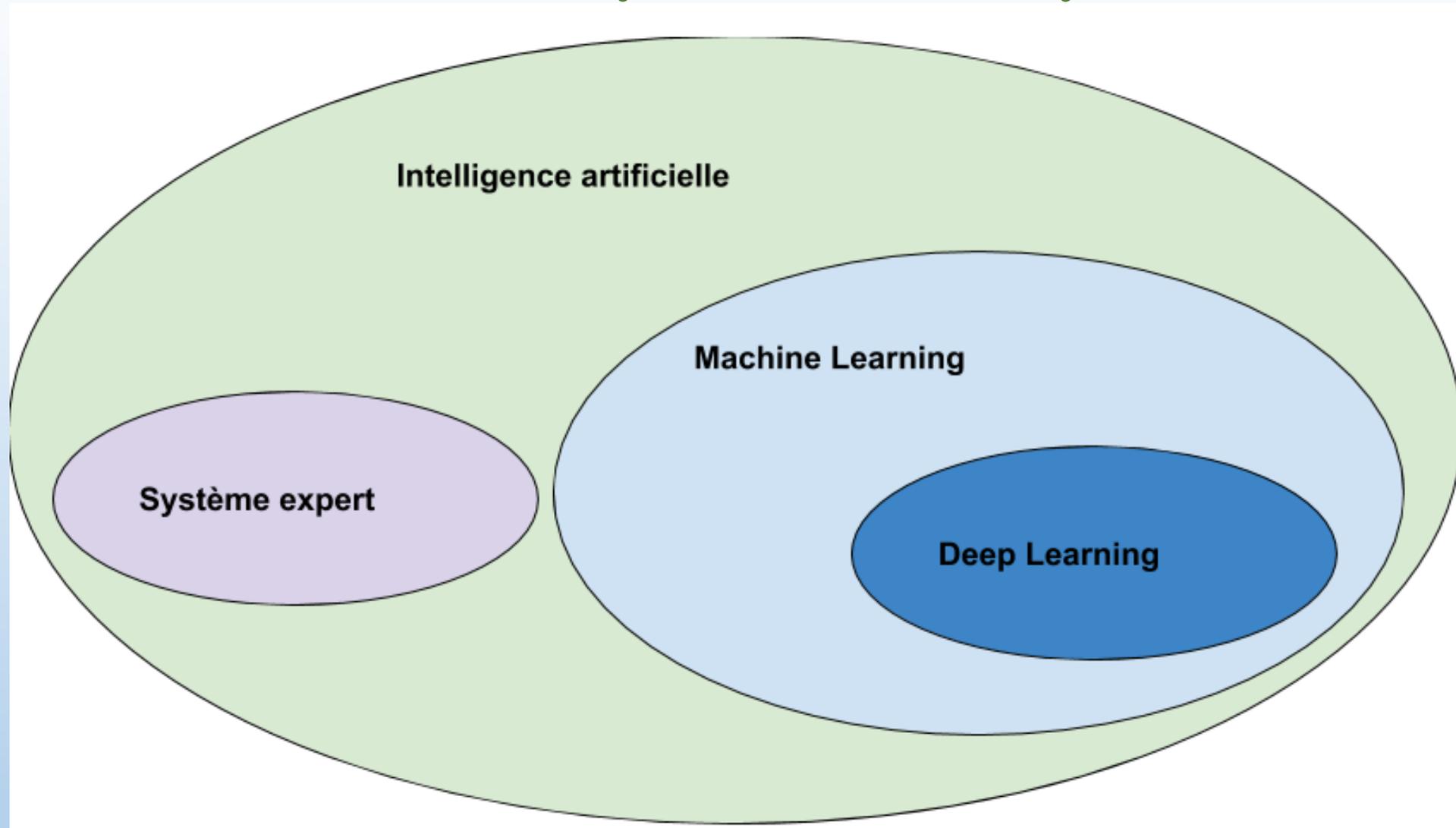
**Cognitivism**



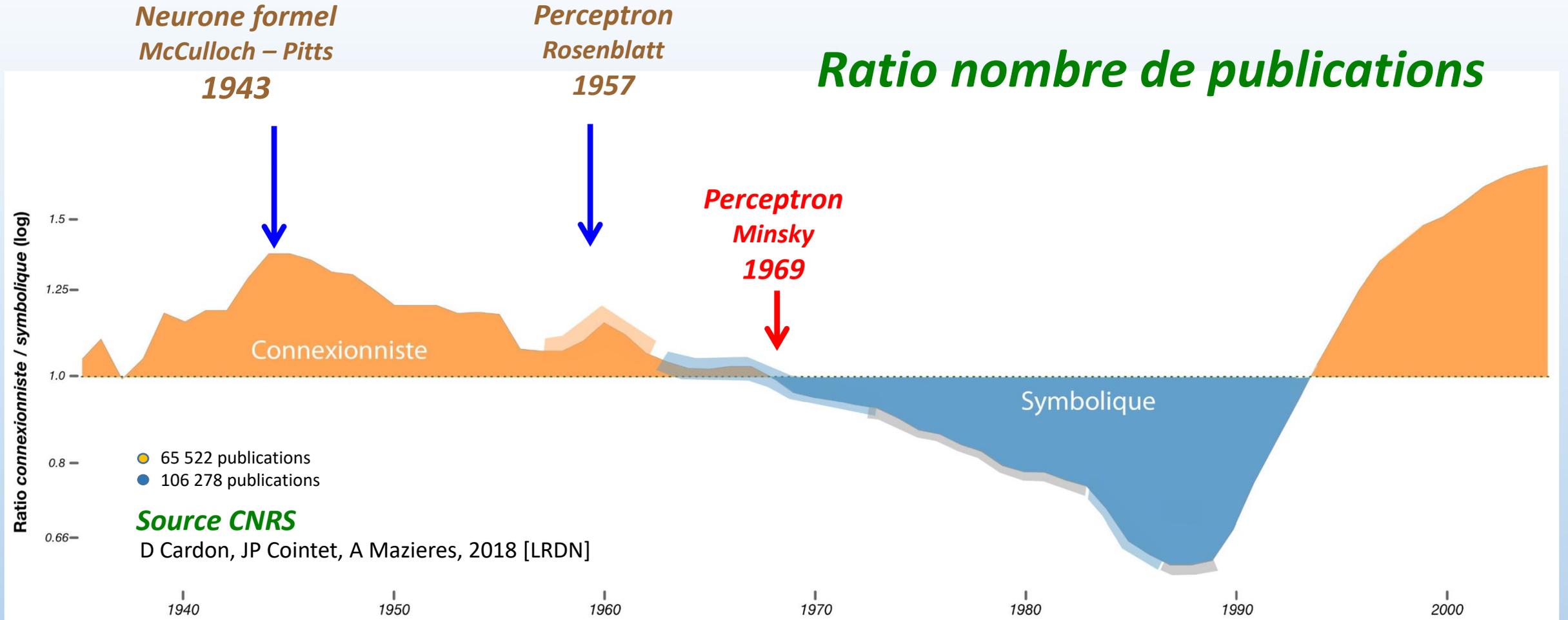
***Sciences humaines***

***Systemes experts***

# Place du *Systeme Expert*



# Approches de l'IA



**1960 1980 : Le connexionnisme** va être marginalisé au profit du **cognitivism**

# Le cognitivisme

**Le cognitivisme** : la pensée peut être décrite à un niveau abstrait comme manipulation de symboles. Cette approche établit un lien entre la pensée et le langage : **système de symboles**.

En 1949, **Warren Weaver** (1894-1978) écrit un mémorandum où il spécule, à la suite **de Leibniz**, sur l'existence de règles linguistiques communes à toutes les langues et veut appliquer ce principe à la traduction automatique sur ordinateur.

Dans le **contexte de la guerre froide**, la traduction automatique du russe en anglais ou de l'anglais au russe paraît cruciale. En 1954, un premier programme, écrit à l'université de Georgetown permet de traduire plusieurs dizaines de phrases simples.

Le programme utilise **250 mots et seulement 6 règles de grammaire** et tourne sur un **IBM 701**.



# Raisonnement symbolique et le théoricien logique

**Simon et Newell** considèrent que la condition *nécessaire et suffisante* pour qu'une machine puisse faire preuve d'intelligence est qu'elle soit un système physique de symboles.

Etre intelligent, c'est aussi être capable de construire des **heuristiques**, de les tester, de les faire évoluer.

Aidés par un programmeur de la **RAND\***, Cliff Shaw, ils développent **Logic Theorist** en **1956**, un programme de démonstration automatique de théorème (voir l'article *The logic theory machine: A Complex Information Processing System*, 1956).

**Logic Theorist** est considéré comme le premier programme informatique relevant du domaine de l'IA.

C'est l'une des premières formulations d'un mouvement philosophique que **John Searle** appellera plus tard « **intelligence artificielle forte** » : comme les humains, les machines peuvent posséder un esprit.

Pour faciliter la programmation du Logic Theorist, Newel, Simon et Shaw développent **le langage IPL (Information Processing Language) en 1956**. Ils inventent pour **cela les structures de listes**.

The Logic Theorist (1956)



\* Research and Development, laboratoire d'idées américain.

## The General Problem Solver (1957)

### General Problem Solver (1957)

#### ➤ Generalization of Logic Theorist

- general framework for solving any problem, given the proper setup
  - rules separated from knowledge base
  - “Weak method” doesn't use domain-specific heuristics
- goal: to be a better psychological model of human cognition
  - have subjects describe what they are doing as they solve problems

#### ➤ Means-ends analysis

- determine differences between current state & goal state
- break into sub-problems [divide & conquer]
- reduce the differences by means of operators

#### ➤ Backtracking

En 1957, **Newell et Simon** programment le **General Problem Solver (GPS)** qui étend les principes utilisés dans le Logic Theorist et implémente ***l'analyse fins moyens*** utilisant **l'arborescence de recherche avec chaînage arrière ou avant**.

GPS fut employé pour résoudre des casse-têtes, des intégrations symboliques, et des décryptages de codes secrets, mais il est surtout resté un programme de recherche

# Les promesses du cognitivisme

Devant ces avancées rapides, **Newell et Simon** prédisent en **1958** dans leur article *Heuristic Problem Solving* **qu'avant 10 ans** :

1. un ordinateur deviendra **champion du monde d'échecs**, sauf si les règles l'empêchent de **participer à la compétition**.
2. un ordinateur **découvrira et prouvera un nouveau et important théorème mathématique**.
3. un ordinateur **écrira de la musique qui sera considérée comme esthétique par les critiques**.
4. la plupart des **théories psychologiques** auront la forme de **programmes d'ordinateur**.

**L'optimisme** est à son comble et les recherches progressent dans toutes les directions.

- Mathématiques : 1961, **James Slagle** (calcul intégrales) -
- Reconnaissance de formes : 1959, **Oliver Selfridge**,
- Détection de similarité 1963 **Tom Evans**,
- Simulation d'un psychothérapeute **ELIZA** **1965 Joseph Weizenbaume**

Au début des années **1960**, le MIT (Minsky), la CMU\* (Newell & Simon) et Stanford (MacCarthy) reçoivent **plusieurs millions de dollars** pour financer les laboratoires d'IA.

\* Carnegie Mellon campus

# 1<sup>er</sup> Hiver 1970 1980

Les grammaires formelles sont **insuffisantes** : ne peuvent traiter de l'ambiguïté (place des mots...), ne prennent pas en compte le contexte etc.

*La découverte de nouveaux théorèmes, prédite par Newell et Simon en 1958 n'est jamais venue*

*Les démonstrations de dialogue machine en langage naturel et de traduction automatique restent hors de portée*

***Et les limites de puissance de calcul et de capacités de stockage :***

*La puissance et la mémoire très insuffisantes : elles suffisaient à peine à démontrer des modèles **simplistes**.*

*Les travaux sur le **raisonnement symbolique** se poursuivent dans le domaine **mathématique***

# Systemes experts

A l'origine du regain d'intérêt et historiquement portés par des langages comme le **Lisp** ou le **Prolog**, les systemes experts ont eu leurs heures de gloire dans des domaines d'applications variées :

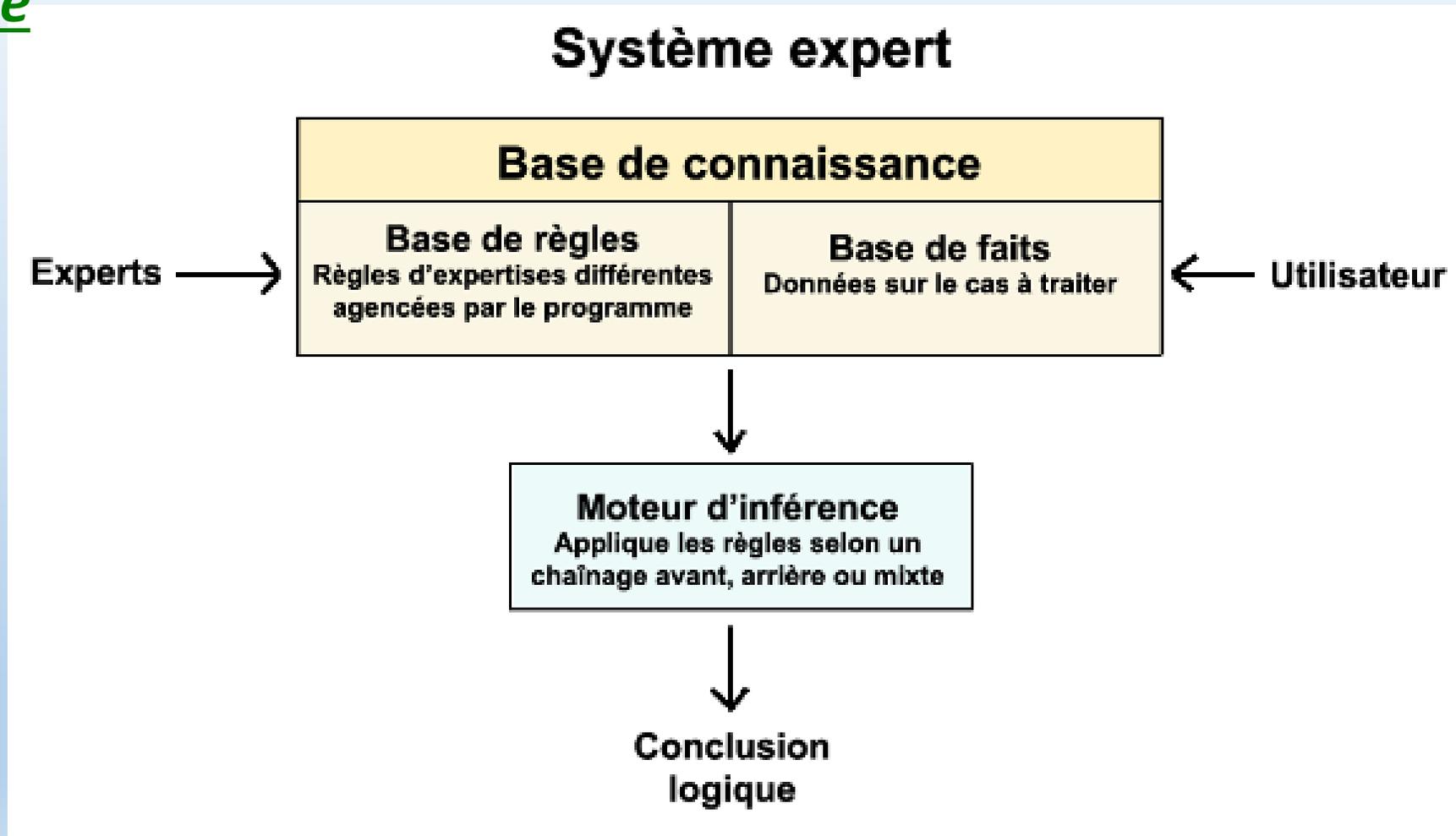
- **DENDRAL**, commencé en **1965**, identifie des composants chimiques à partir de relevés spectrométriques
- **MYCIN en 1972**, le plus connu des SE, est dédié au diagnostic de maladies infectieuses du sang. En 1978, **MYCIN, avec ses 500 règles**, obtient score plus élevé que des experts humains.

Aujourd'hui, dans le domaines de l'intelligence artificielle, les systemes experts ont moins la cote et sont plutôt employés dans **l'aide décisionnelle**

# Systemes experts

Branche « déterministe » de l'IA

Conçu pour simuler le savoir-faire d'un spécialiste, dans un domaine précis et bien délimité



# Systemes experts

L'ensemble **modèle de faits et base de règles** est aussi appelé la **base de connaissance**.

Le **moteur d'inférence** utilise **faits et règles** pour produire de nouveaux **faits**, jusqu'à parvenir à un objectif ou à la réponse à la question posée initialement.

La **définition des règles** pertinentes dépend des algorithmes de résolution inclus dans le moteur d'inférence.

Les plus couramment utilisés sont :

- le chaînage avant, partant des faits pour en rechercher les conséquences,
- le chaînage arrière, partant des conclusions envisagées et vérifie si les conditions sont réunies

**Difficulté** : acquisition des connaissances d'un expert et de leur formalisation en tant que faits (données manipulées) et règles.



# Systemes experts

*Exemple de règle de **DENDRAL** :*

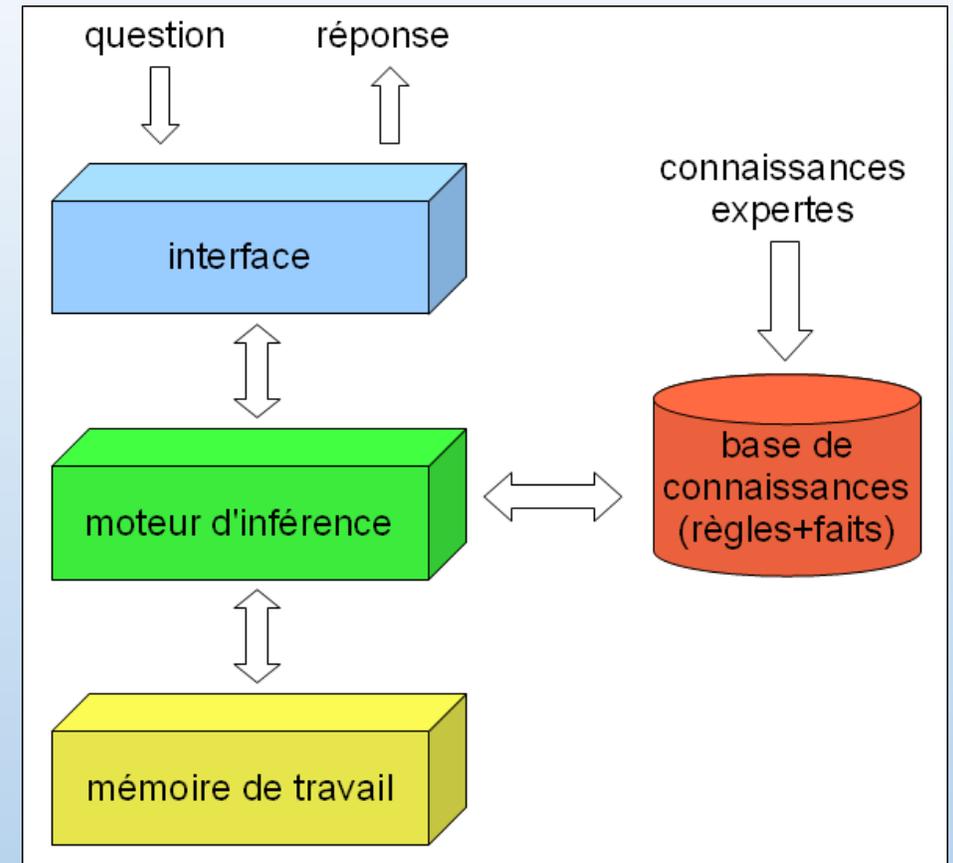
***Si** le spectre de la molécule présente deux pics  $x_1$  et  $x_2$   
tels que  $x_1 - x_2 = M + 28$*

***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*



# Systemes experts - Points forts

*L'encapsulation (ou la représentation) de la connaissance sous la forme de règles logiques permet la maintenance de la base de règles,*

*Cela favorise la transmission du savoir ou l'extension à d'autres domaines (par l'ajout de règles plus spécifiques si nécessaire).*

*La mise au point est rapide et est d'une grande fiabilité.*

*Le système expert est maîtrisé par ses concepteurs.*

*Les systèmes experts peuvent raisonner sur des données **partiellement connues** avec de la **logique floue**, c'est-à-dire des connaissances approximatives ou incertaines comme « la température du four est très élevée », « l'enfant est grand », « le contrat est important » ...en utilisant des coefficients de vraisemblance*



# Systemes Experts - Limitations

*Un expert humain est nécessaire. Il doit pouvoir coopérer au projet et pouvoir expliquer son raisonnement alors que de nombreux experts éprouvent des difficultés à les expliciter car le savoir métier est peut-être perdu ou non documenté.*

*Le domaine d'expertise du système expert est très étroit et est limité aux frontières du domaine métier.*

*Les systèmes experts, contrairement aux technologies plus récentes de l'IA (Machine Learning) ne permettent pas l'apprentissage et ne gèrent pas la reconnaissance d'images ou de la parole, ce qui limite leurs portées.*



# *Systemes formels*

# Systemes formels - Définition

**Un système formel** est « un ensemble de données purement abstrait, sans lien avec l'extérieur, qui décrit les règles de manipulation d'un ensemble de symboles traités de façon uniquement syntaxique, c'est-à-dire sans considération de sens (sémantique).

Il est constitué :

1°/ **d'un alphabet fini** de symboles;

2°/ **d'un procédé de construction** des mots du système formel;

3°/ d'un ensemble **d'axiomes qui sont des mots**;

4°/ d'un ensemble fini de **règles de déduction** qui permettent de déduire d'un ensemble fini de mots un autre ensemble de mots.

Elles sont de la forme :

$U_1 \text{ et } U_2 \text{ et } \dots U_p \implies W_1 \text{ et } W_2 \text{ et } \dots W_n$

où les  $U_i$  et les  $W_j$  sont des mots du système formel.

La flèche " $\implies$ " se lit "permet de déduire". »

# Exemple du langage naturel

L'exemple suivant est emprunté à Chomsky. Il décrit une **grammaire élémentaire** basée sur six règles de réécriture :

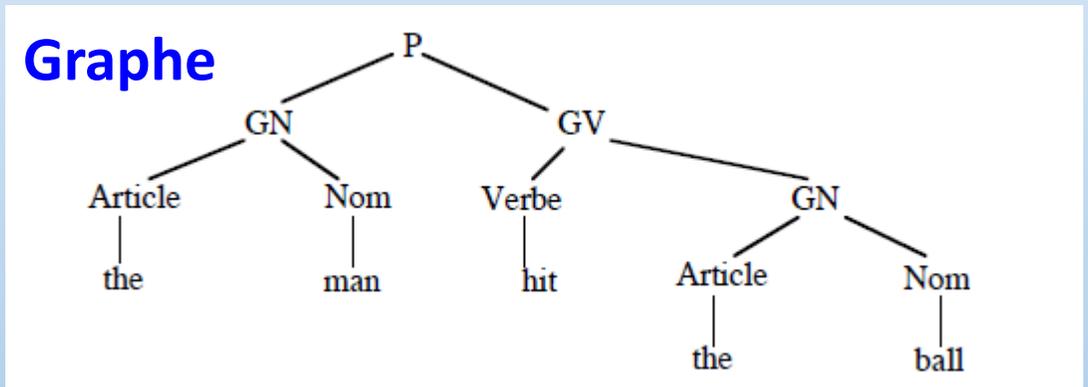
- (I)  $P \rightarrow GN+GV$
- (II)  $GN \rightarrow Article+Nom$
- (III)  $GV \rightarrow Verbe+GN$
- (IV)  $Article \rightarrow the$
- (V)  $Nom \rightarrow man, ball$
- (VI)  $Verbe \rightarrow hit, took$

**Le langage est compositionnel**

**Le caractère additif d'un langage objets :**  
**il suffit d'ajouter de nouvelles clauses (sans modifier les précédentes) pour "apprendre" au programme de nouveaux mots ou de nouvelles règles de grammaire.**

Par applications successives de ces règles, on peut aboutir à la décomposition suivante :

P  
GN+GV (I)  
Art+Nom+GN (II)  
Art+Nom+Verbe+GN (III)  
Art+Nom+Verbe+Article+Nom (II)  
the+Nom+Verbe+the+Nom (IV)  
the+man+Verbe+the+ball (V)  
the+man+hit+the+ball (VI)



# Systemes formels - Exemple du jeu d'échecs

L'ensemble des parties du jeu d'échecs pourrait ainsi être représenté par le **systeme formel** :

- **Un alphabet** : tout symbole représentant une pièce du jeu positionnée sur l'échiquier (toutes les pièces possibles, sur toutes les cases possibles).

Exemple: "Fou Noir en A1", " Cavalier Blanc en B5", etc ...

- **Des mots** : toute combinaison finie de pièces positionnées. Toute configuration possible de l'échiquier peut donc être représentée par un mot (les configurations "absurdes" comprenant par exemple plusieurs pièces sur la même case sont également des mots),

-**Un axiome** : mot représentant la configuration initiale de l'échiquier.

-**Des règles de déduction** : les règles de mouvement des pièces sur l'échiquier.

Autrement dit, comment partant d'une configuration du jeu (donc d'un mot) déplacer une pièce (donc créer un autre mot).

# Systemes formels - Exemple du jeu d'échecs (suite)

**L'arbre de dérivation** présentera **toutes les parties possibles du jeu, se terminant** (extrémité de chaque branche de l'arbre), soit par un « mat », soit par un « pat ».

**L'automate joueur d'échec anticipera, dans la limite de sa mémoire, n coups à l'avance, en simulant alternativement**

- le mouvement qui lui est le plus favorable, lorsque c'est à lui de jouer,
- Le mouvement le plus défavorable, lorsque c'est au tour de l'adversaire

**D'où le nom de cette méthode, dite "min-max".**

**Pour évaluer les situations favorables ou défavorables, autrement dit limiter l'exploration de l'arborescence en éliminant les branches "sans intérêt", l'automate s'appuiera sur la pratique d'un expert.**

# Prolog

Le langage **Prolog (PROgrammation LOGique)** a été conçu par Alain Colmerauer à Marseille en **1970** pour le domaine de la linguistique.

Un programme Prolog est constitué de clauses. Celles-ci sont de trois types : **faits, règles et questions.**

- Une relation peut être spécifiée par des faits, en énonçant simplement les  $n$ -uplets\* des objets vérifiant la relation, ou en établissant des règles concernant la relation.
- Une procédure est un ensemble de clauses concernant une même relation

Par exemple, le fait élémentaire : **Pierre achète une voiture** sera représenté par une clause : **acheter (Pierre,Voiture)**

Qu'on peut généraliser par : **acheter ( X , Y )** X et Y étant des objets.

-\*  $n$ -uplet = collection ordonnée de  $n$  objets

# Exemples de Règles

## Prolog

Connaissant, par exemple, les faits élémentaires :

Père ( Jean , Paul ).      (pour "Jean est le père de Paul" )

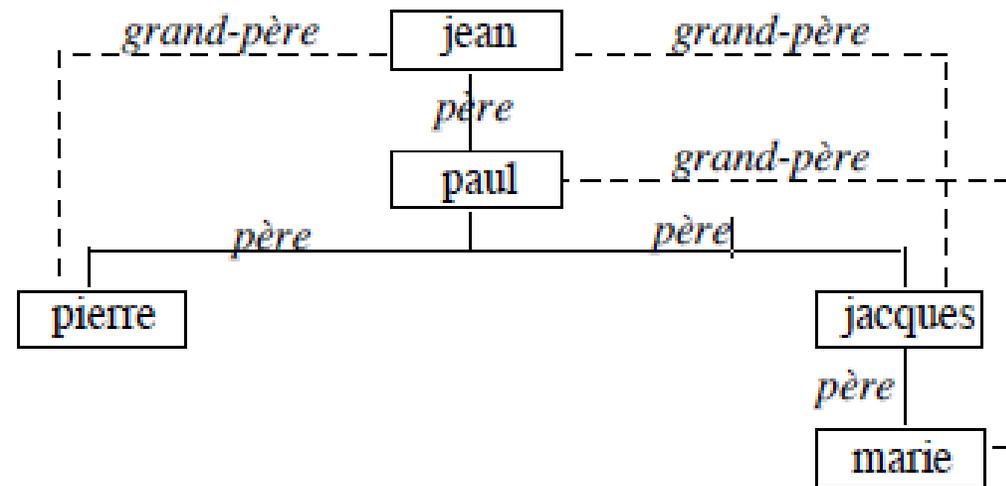
Père ( Paul , Pierre).

Père ( Paul , Jacques ).

Père ( Jacques, Marie).

La règle "Grand-Père", permettra de produire (**d'inférer**) de nouveaux faits :

Grand-Père ( X , Y ) :-  
  il existe un objet Z avec  
  Père ( X , Z ), et  
  Père ( Z , Y ) .



# Exemples de Règles

## Prolog

Avec la base de faits précédente, l'interrogation :

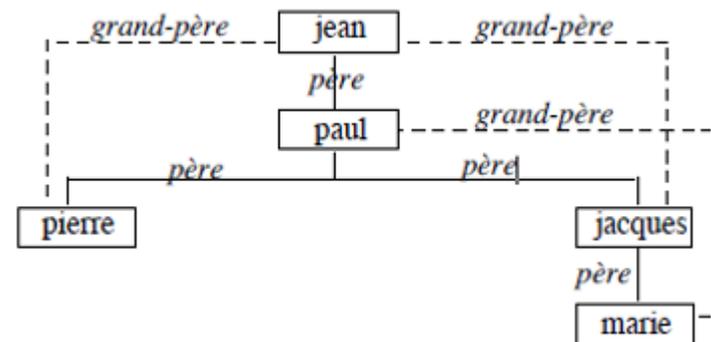
Père ( Jean , Pierre ) ?  
a pour réponse "faux".

(i.e. "Jean est-il le père de Pierre ?")

Et l'interrogation :

Grand-Père ( Jean , X ) ?  
a pour réponse :  
X = Pierre, et  
Y = Jacques.

(i.e. "Jean est le grand-père de qui ?")

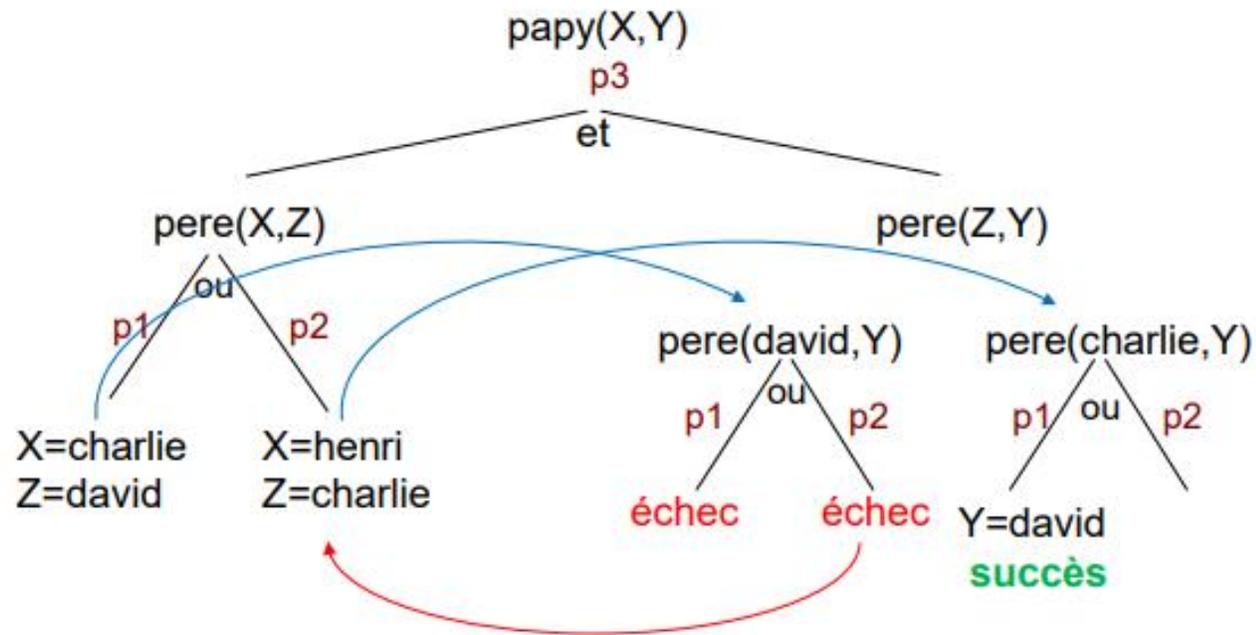


# Prolog Règles

Exemple de programme:

```
pere(charlie, david).      (p1)  
pere(henri, charlie).    (p2)  
papy(X,Y) :- pere(X,Z), pere(Z,Y). (p3)
```

Appel du programme : papy(X,Y).



# Exemple

## Exploration arbre

planete(X) :- astre(X), etoile(Y), satellite(X,Y).

faits:

astre(lune).

astre(terre).

astre(soleil).

astre(venus).

satellite(venus, soleil).

satellite(lune, terre).

satellite(terre, soleil).

étoile(vega).

étoile(soleil).

?- planete(X).

# Exemple Exploration arbre

Planete (X) :- astre (X), etoile (Y), satellite (X,Y)

planete(X) :- astre(terre), etoile(vega), satellite(terre, vega).

faits: échec !

astre(lune).

astre(terre).

astre(soleil).

astre(venus).

satellite(venus, soleil).

satellite(lune, terre).

satellite(terre, soleil).

étoile(vega).

étoile(soleil).

planete(X) :- astre(terre), etoile(soleil), satellite(terre, soleil).

faits: backtrack true !

astre(lune).

astre(terre).

astre(soleil).

astre(venus).

satellite(venus, soleil).

satellite(lune, terre).

satellite(terre, soleil).

étoile(vega).

étoile(soleil).

X = terre,

Y = soleil

# Table de décision

*Beaucoup de règles peuvent être décrites sous la forme d'un tableau : **table de décision.***

*Elles sont idéales lorsqu'il existe des règles similaires avec des valeurs différentes.*

- *chaque colonne = condition ou action,*
- *chaque ligne = règle distincte.*

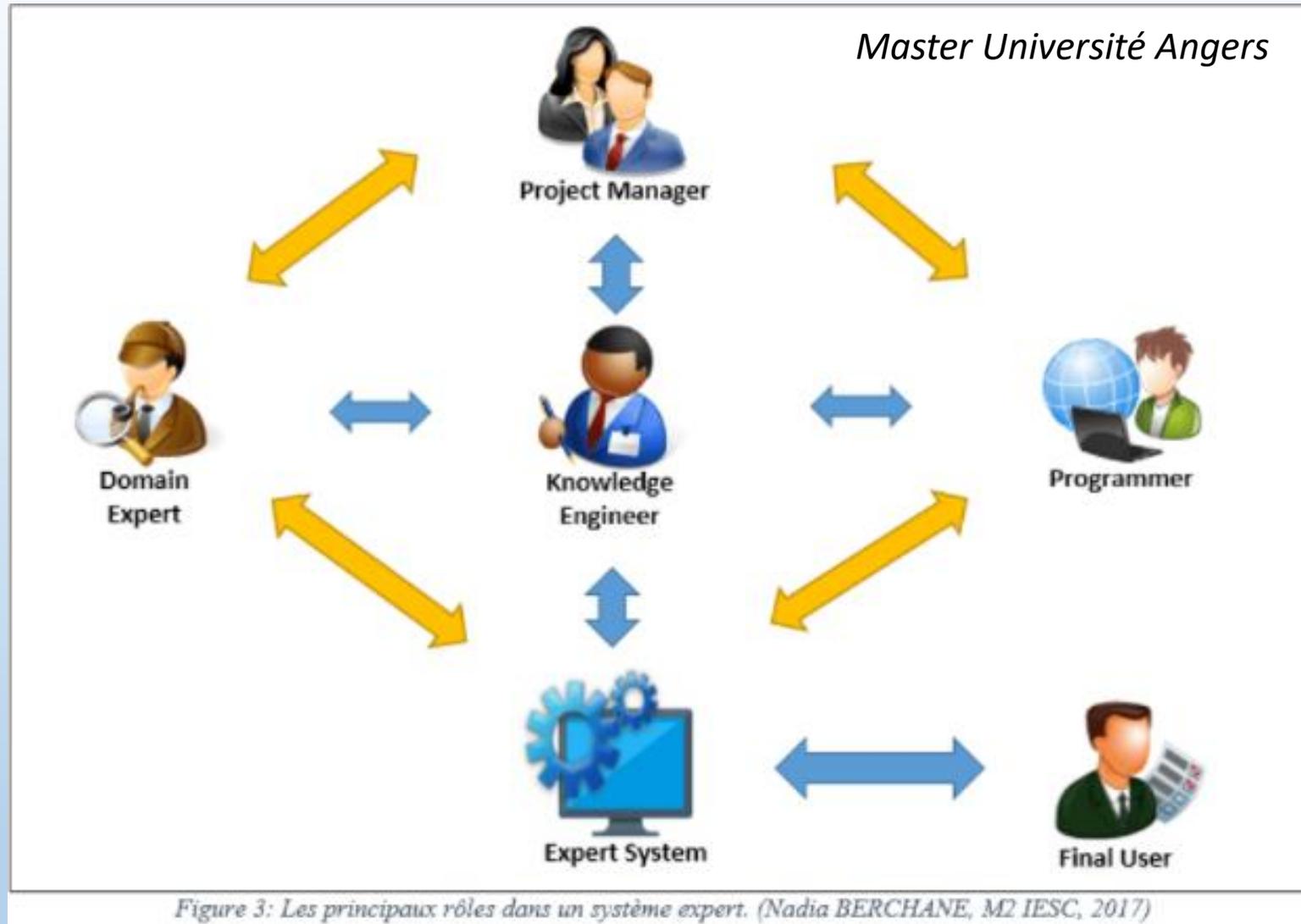
*Cette vision est synthétique et compréhensible par tous les acteurs du projet (expert métier, cognicien, IT...).*

Table des catégories de risque					
U	Client existant	Taux du risque de la demande	Score Crédit	Catégorie du risque	
1	NON	<120	< 590	FORT	
2			[590..610]	MOYEN	
3			>610	FAIBLE	
4		[120..130]	>130	< 600	FORT
5				[600..625]	MOYEN
6				>625	FAIBLE
7				-	TRES FAIBLE
8	OUI	<= 100	< 580	FORT	
9			[580..600]	MOYEN	
10			>600	FAIBLE	
11		>100	< 590	FORT	
12			[590..615]	MOYEN	
13			>615	FAIBLE	

Exemple de table de décision utilisé en DMN (Decision Model Notation)

Risque crédit

# Les acteurs dans un Système Expert



# La probabilisation

*Le volume croissant des connaissances en entrée et la complexification des réseaux de concepts → les règles de raisonnement deviennent conditionnelles et peuvent être **probabilisées**.*

*En traitant des **heuristiques** plutôt que de la logique, la catégorie **vrai/faux** est moins importante que **fructueux/stérile**.(Minsky )*

*Parmi les milliers de règles formulées par les experts, il est possible à partir d'une prémisse fixée (**SI...**), d'établir une **probabilité relative** au fait que la deuxième proposition (**ALORS...**) a un **pourcentage de chance d'être vraie**.*

# La probabilisation

*Les méthodes statistiques pour éliminer les hypothèses candidates vont ainsi mûrir et se développer en produisant des raisonnements de type inférentiel comme :*

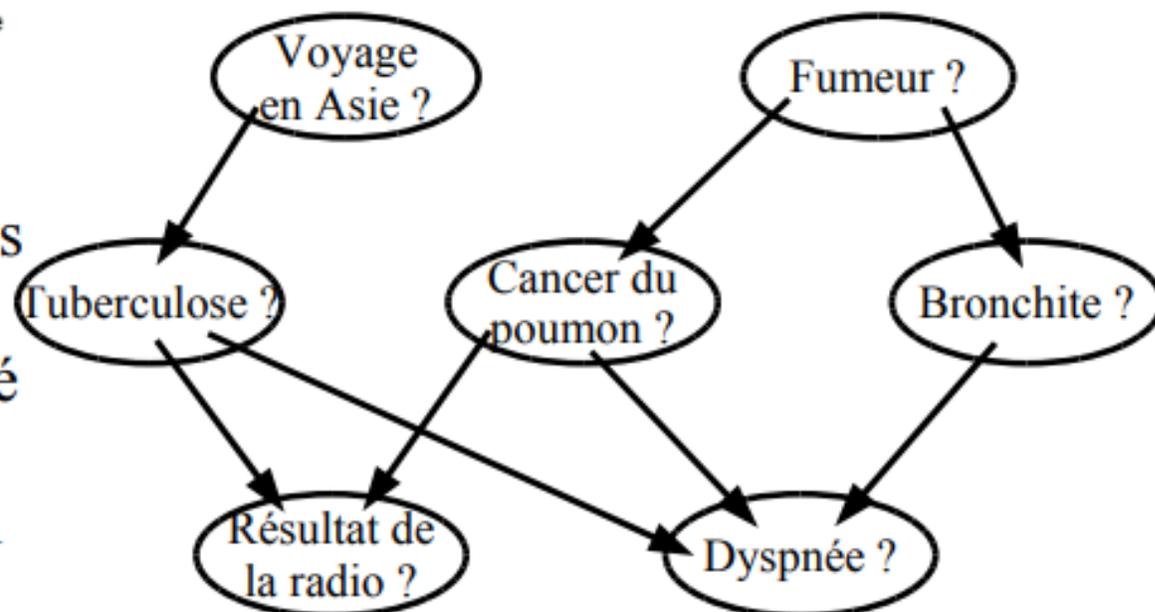
- *les arbres de décision (qui, par la suite, donneront naissance aux forêts aléatoires (random forests),*
- *les techniques dites « divide and conquer »,*
- *les réseaux bayésiens \* qui permettent d'ordonner des dépendances entre variables avec un formalisme causaliste*

Domingos, 2015

\* Un réseau bayésien, graphe orienté, est à la fois :

- un modèle de représentation des connaissances ;
- une « machine à calculer » des probabilités conditionnelles
- une base pour des systèmes d'aide à la décision<sup>1</sup>

**Objectif** : produire des prédictions sur les caractéristiques non observées d'une entité en utilisant les propriétés locales qui lient ces caractéristiques.



The Asian network  
(Lauritzen *et al.*, 1988)

Un réseau bayésien est un **ensemble de variables aléatoires** respectant une structure de **dépendances/indépendances conditionnelles** traduite par un **graphe acyclique dirigé**.

A chaque variable aléatoire est attachée la distribution de probabilité marginale ou conditionnelle (suivant son statut) dans le graphe dont le produit fournit la distribution de **probabilité conjointe de l'ensemble**.

Lois locales (liens avec les parents) :

$$P(A=\text{oui})=0,01$$

$$P(T=\text{oui}/A=\text{oui})=0,05$$

$$P(T=\text{oui}/A=\text{non})=0,01$$

$$P(F=\text{oui})=0,5$$

$$P(C=\text{oui}/F=\text{oui})=0,1$$

$$P(C=\text{oui}/F=\text{non})=0,01$$

$$P(B=\text{oui}/F=\text{oui})=0,6$$

$$P(B=\text{oui}/F=\text{non})=0,3$$

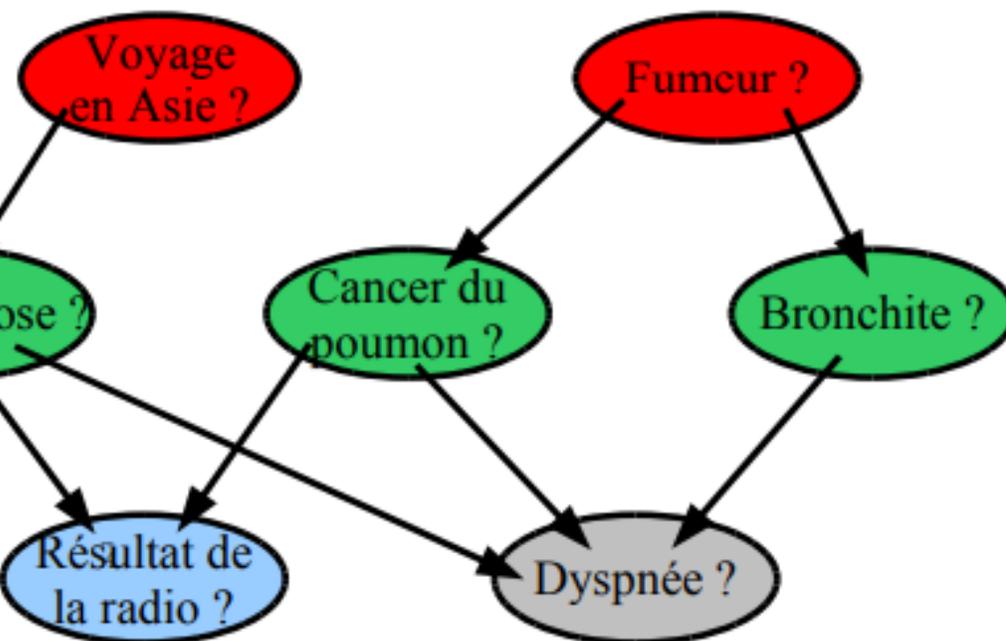
$$P(R=\text{oui}/T=\text{oui},C=\text{oui})=0,98$$

$$P(R=\text{oui}/T=\text{oui},C=\text{non})=0,05$$

$$P(R=\text{oui}/T=\text{non},C=\text{oui})=0,05$$

$$P(R=\text{oui}/T=\text{non},C=\text{non})=0$$

$$P(D=\text{oui}/B=\text{oui},C=\text{oui},T=\text{oui})=0,9$$



$$P(D=\text{oui}/B=\text{non},C=\text{oui},T=\text{non})=0,8$$

$$P(D=\text{oui}/B=\text{non},C=\text{oui},T=\text{oui})=0,7$$

$$P(D=\text{oui}/B=\text{oui},C=\text{non},T=\text{non})=0,8$$

$$P(D=\text{oui}/B=\text{non},C=\text{non},T=\text{non})=0,1$$

$$P(D=\text{oui}/B=\text{oui},C=\text{oui},T=\text{non})=0,9$$

$$P(D=\text{oui}/B=\text{non},C=\text{non},T=\text{oui})=0,8$$

$$P(D=\text{oui}/B=\text{oui},C=\text{non},T=\text{oui})=0,9$$

# Types de systèmes experts

*Haton & Haton (1989) distinguent différents types de systèmes experts :*

## ***Systèmes d'interprétation de données.***

*Exemples :*

- Systèmes de diagnostic en médecine ("de quelle maladie s'agit-il?"),*
- Système d'interprétation géologique ("les mesures séismologiques permettent-elles de croire à l'existence de dépôts minéraux importants?"),*
- Systèmes d'évaluation psychologique ("s'agit-il d'un cas suicidaire?"), etc.*

## ***Systèmes de prédiction.***

*Exemples :*

- Systèmes de prédiction météorologique ("Il pleut aujourd'hui en France. va-t-il pleuvoir en Suisse demain?"),*
- Prédiction géopolitiques ("Les conflits de guerre sont particulièrement fréquents en situation de crise économique.*

# Types de systèmes experts

## **Systemes de planification.**

*Exemples :*

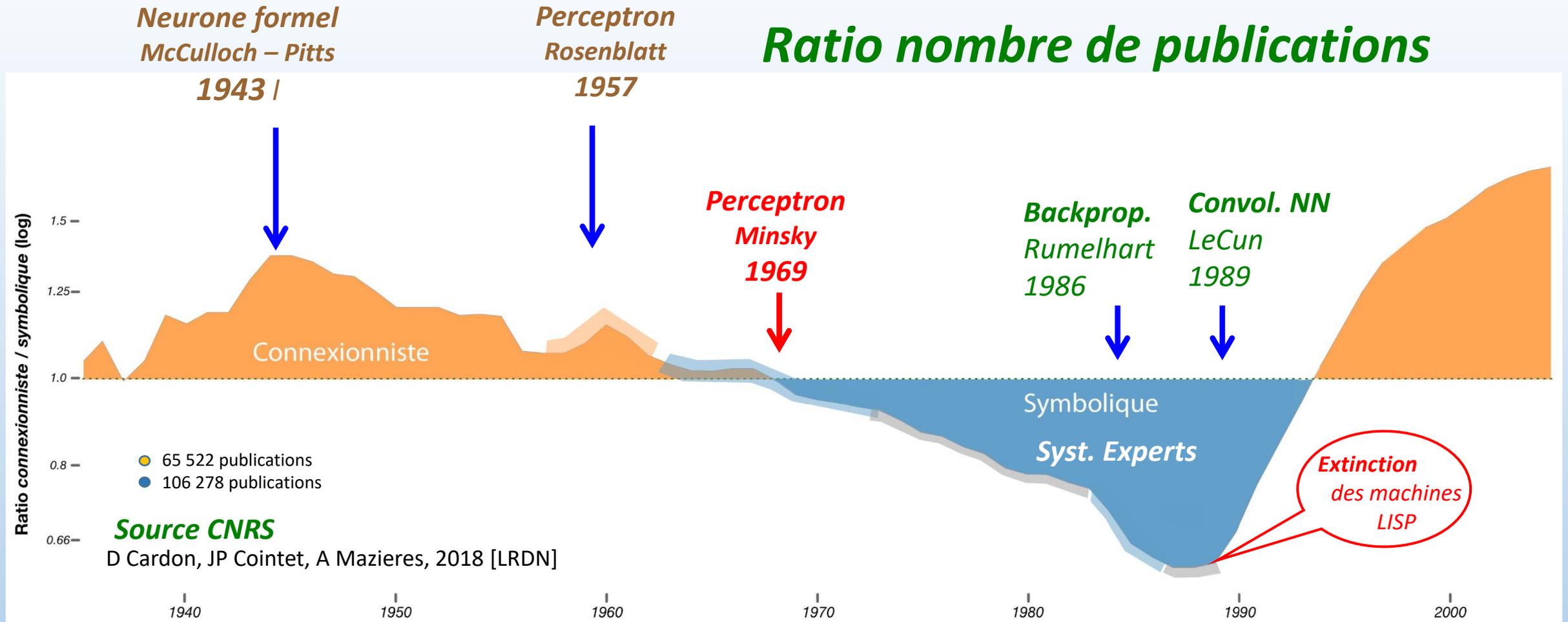
- *Systeme de reservation de vols aeriens, planification des altitudes de vol selon les vents connus et les corridors disponibles,*
- *Planification des actions d'assemblage d'un robot industriel,*
- *Planification des interventions requises pour la construction d'un bdtiment, etc.*

## **Systemes de conception.**

*Exemples :*

- *Dveloppement et simplification de circuits intgrs,*
- *Amnagement d'une cuisine optimale dans un espace donn,*
- *Clonage de gnes, cration d'un nouveau compos chimique, etc.*

# Fin des années 80 Extinction des machines LISP



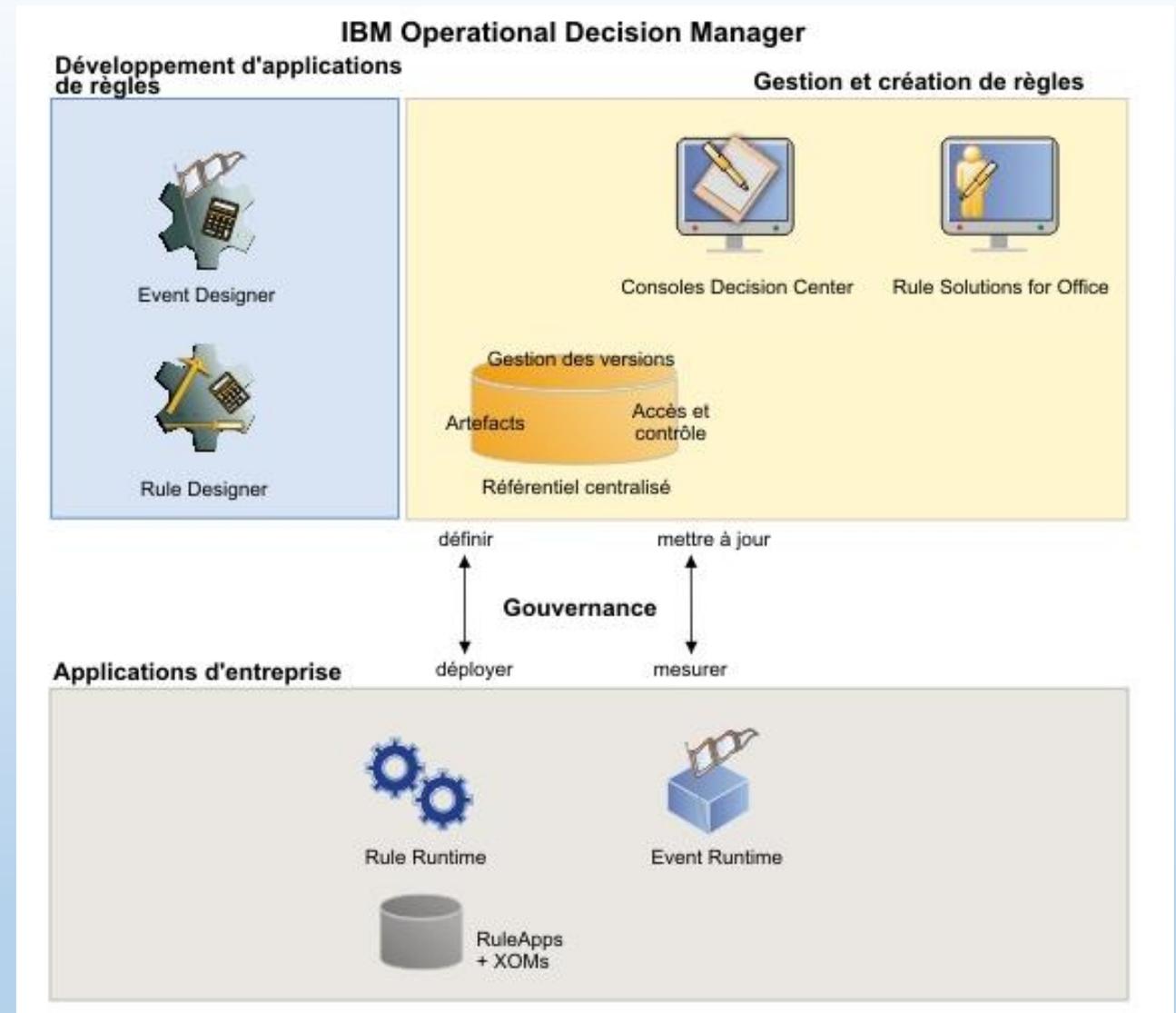


# IBM ODM

**Operational Decision Manager** comprend :

*Decision Server pour la gestion des décisions et la détection des événements*

*Decision Center pour mettre la gestion des décisions dans les mains de ceux qui pilotent l'activité*



# Drools

**Drools** est un  **système expert** ou  **Business Rules Management System (BRMS)**.

Il utilise un  **moteur de règles** [permettant de déclencher des  **actions** sur la base d'un  **ensemble de règles** si des données les valident ou non.

**Un système expert** tel que Drools trouve son intérêt dans les systèmes logiciels où il est nécessaire de réaliser de nombreux tests sur des données entrantes afin d'en produire d'autres déduites des premières apportant plus de  **valeur** pour par exemple  **une prise de décision**.

Une règle se définit tout simplement par la formule :

```
IF [test] THEN [actions]
```

**DMN**, qui signifie  **Decision Model and Notation**, est une norme relativement nouvelle (2015) gérée par  **Object Management Group (OMG)**, l'organisation derrière  **BPMN (Business Process Model Notation)** et  **UML (Unified Modelling language)**. Elle fait la promotion de  **DMN** pour décrire les règles dans un formalisme indépendant d'une solution logicielle et d'un éditeur.

**Drools** est le premier environnement d'exécution  **DMN** open-source qui prend en charge le niveau de conformité complet de la spécification  **DMN**.



# Drools Exemple

Golfer attributs :

name : String  
position : int  
color : String

*Il y a quatre golfeurs alignés de gauche à droite.*

*Joe est deuxième dans la ligne ;*

*Bob porte un pantalon à carreaux ;*

*Tom n'est pas en position une ou en position quatre,  
et il ne porte pas un pantalon orange;*

*Le golfeur à droite immédiate de Fred porte un  
pantalon bleu ;*

when

```
// il existe un golfeur nommé Fred
fred : Golfer ( name == "Fred" )

// Joe est en position 2
joe : Golfer ( name == "Joe",
              position == 2 && != fred.position,
              color != fred.color )

// Bob est habillé d'un pantalon à carreau
bob : Golfer ( name == "Bob",
              position != fred.position && != joe.position,
              color == "carreau" && != fred.color && != joe.color )

// Tom n'est pas en position une ou en position quatre,
// et il ne porte pas un pantalon orange
tom : Golfer ( name == "Tom",
              position not in (1, 4, fred.position, joe.position, bob.position),
              color not in ("orange", fred.color, joe.color, bob.color))

// Le golfeur à droite immédiate de Fred porte un pantalon bleu
Golfer ( position == (fred.getPosition() + 1),
        color == "bleu",
        this in ( joe, bob, tom ))
```

***Donnez la liste des joueurs et leurs caractéristiques : nom, position et couleur de pantalon.***



# Drools

La partie « action » de cette règle contiendra l'affichage du résultat sur la console :

```
then
  System.out.println("La position de Fred : " + fred.getPosition()+ ", sa couleur est " + fred.getColor());
  System.out.println("La position de Joe : " + joe.getPosition() + ", sa couleur est " + joe.getColor());
  System.out.println("La position de Tom : " + tom.getPosition() + ", sa couleur est " + tom.getColor());
  System.out.println("La position de Bob : " + bob.getPosition() + ", sa couleur est " + bob.getColor());
end
```

Résultats d'exécution :

- La position de Fred : 1, sa couleur est orange
- La position de Joe : 2, sa couleur est bleue
- La position de Tom : 3, sa couleur est rouge
- La position de Bob : 4, sa couleur est carreau



# Drools

## Optimiseur de ressources Opta Planner

Which example do you want to see?

Basic examples	Real examples	Difficult examples
 N queens	 Course timetabling	 Exam timetabling
 Cloud balancing	 Machine reassignment	 Employee rostering
 Traveling salesman	 Vehicle routing	 Traveling tournament
 Dinner party	 Project job scheduling	 Cheap time scheduling
 Tennis club scheduling	 Hospital bed planning	 Investment asset class allocation

*Planification des ressources commerciales, optimisation de trajets de véhicules de livraison, gestion du stationnement d'une flotte automobile, planning des employés, planning des cours pour les élèves et professeurs, optimisation de ressources dans le cloud, attribution de tâches, planification de travaux...*



# Diffusion dans les entreprises

## Rôle important dans la compétitivité des entreprises et l'intelligence économique

« L'objectif de l'établissement d'une **mémoire d'entreprise** en se servant des **systemes experts** est d'assister l'utilisateur, en lui fournissant des informations pertinentes sur l'entreprise, et des recommandations basées sur des retours d'expérience déjà capitalisés sur la base de données du système expert, tout en lui laissant la responsabilité d'interpréter et évaluer l'information retournée par le système.

Tout cela permet une meilleure exploitation des ressources informationnelles de l'entreprise, et par la suite une optimisation **de sa performance.** »

Master Université Angers 2017



# Assistants personnels (intelligents)



Les chatbots comme **Siri d'Apple ou Google Now, en général des** systèmes experts. 2010.

Nourris par des experts humains.

Arbre de décision.

Commande et synthèse vocales .

## Exemples d'applications Fonctions

- Effectuer un appel ;
- Dictier un SMS à envoyer ;
- Dictier une recherche à faire sur le web
- Jouer un morceau de musique
- Lancer un itinéraire;
- Chercher des restaurants ou encore des cinémas.
- En cuisine
- Faire des calculs
- Discuter.: limitations
- ....



ITW Luc Julia 6'

<https://www.youtube.com/watch?v=fC3pQfD0W1A>

# Watson

***Watson** est un programme informatique d'intelligence artificielle conçu par la société IBM dans le but de répondre à des questions formulées en langage naturel.*

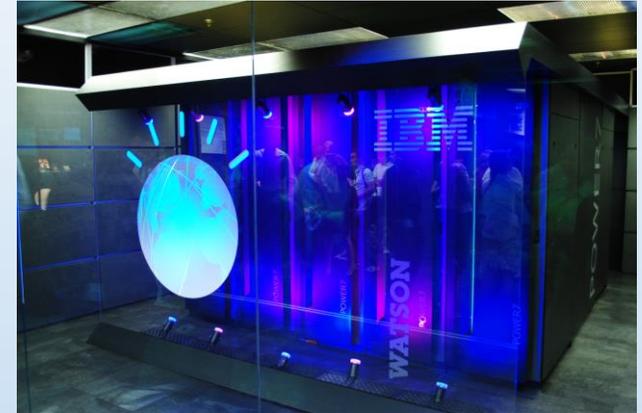
*Il s'intègre dans un programme de développement plus vaste, le DeepQA research project.*

***Le nom « Watson »** fait référence à Thomas J. Watson, dirigeant d'IBM de 1914 à 1956, avant même que cette société ne s'appelle ainsi.*

***En 2011**, **Watson** connaît une notoriété au niveau mondial quand il devient **le champion du jeu télévisé américain Jeopardy!**, en battant les meilleurs concurrents humains de l'histoire de ce jeu.*

# Watson

- **90 serveurs Power 750** réunis dans dix racks.
  - Chaque Power750 contient 4 processeurs octo-cores/ 3,55 GHz, soit 32 cœurs par serveur, soit au total **2 880 cœurs Power7**. 10 racks
  - **mémoire vive : 16 téraoctets**
- **puissance de calcul : 80 téraflops**



*Réponses aux questions dans des délais comparables à ceux de ses concurrents humains*

*Parcours d'une grande quantité de contenu disponible en local : **deux cents millions de pages** en moins de **trois secondes***

# Watson

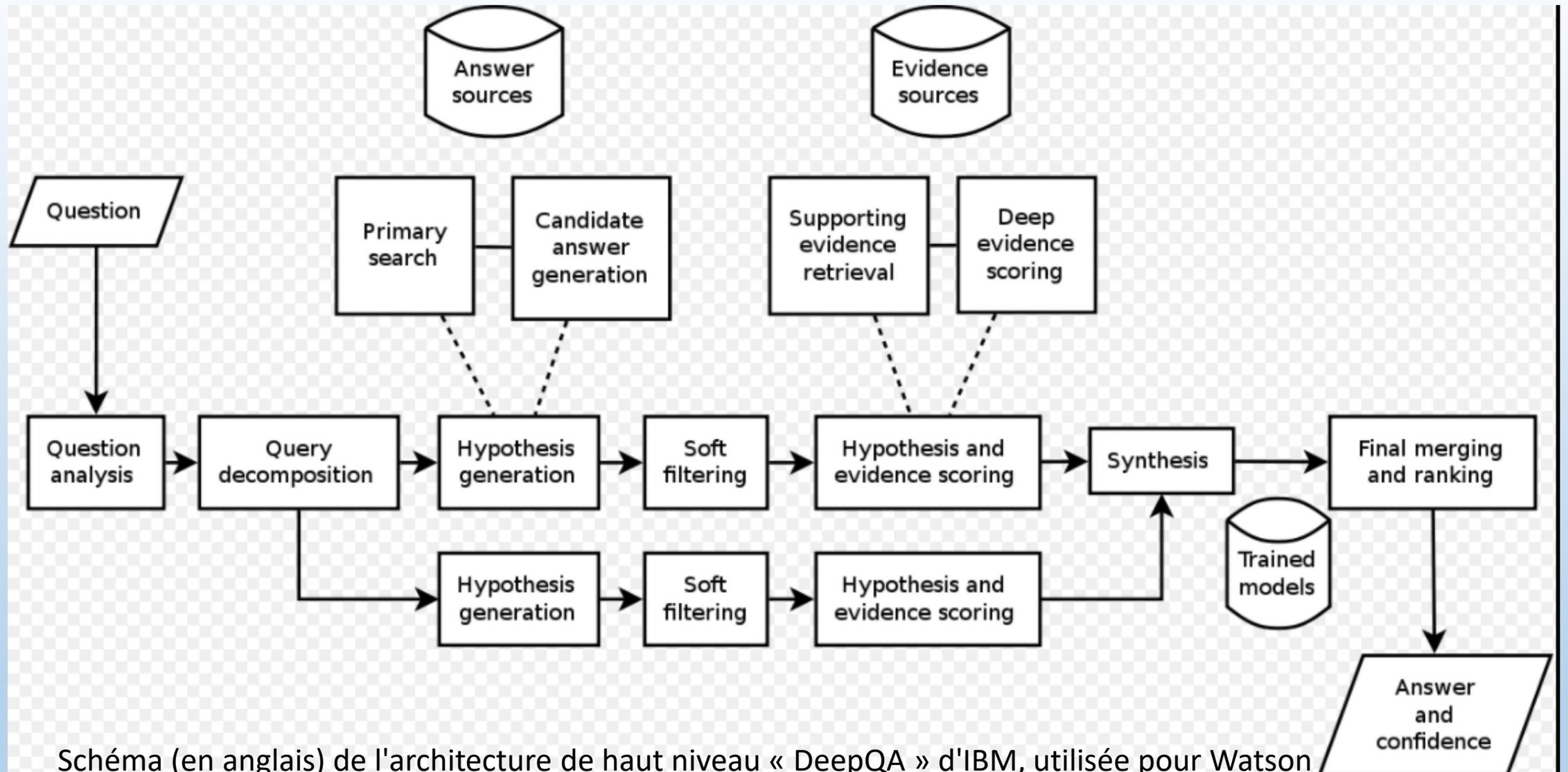


Schéma (en anglais) de l'architecture de haut niveau « DeepQA » d'IBM, utilisée pour Watson

# Watson

Comment Watson répond à une question ?



# Watson

*En 2016, un centre de recherche de la faculté de médecine de l'université de Tokyo dirigé par le professeur Arinobu Tojo a utilisé Watson pour aider les **médecins** dans le **diagnostic** d'un cas rare de leucémie.*

*En dix minutes, le supercalculateur a trouvé la véritable cause ; les médecins estiment qu'il leur aurait fallu deux semaines pour faire la même tâche, rendant le traitement beaucoup plus incertain.*

***L'assistant juridique baptisé Ross** (basé sur l'API Watson), capable d'analyser la jurisprudence dans certains domaines, a été acheté par un cabinet d'avocats américain en 2016.*

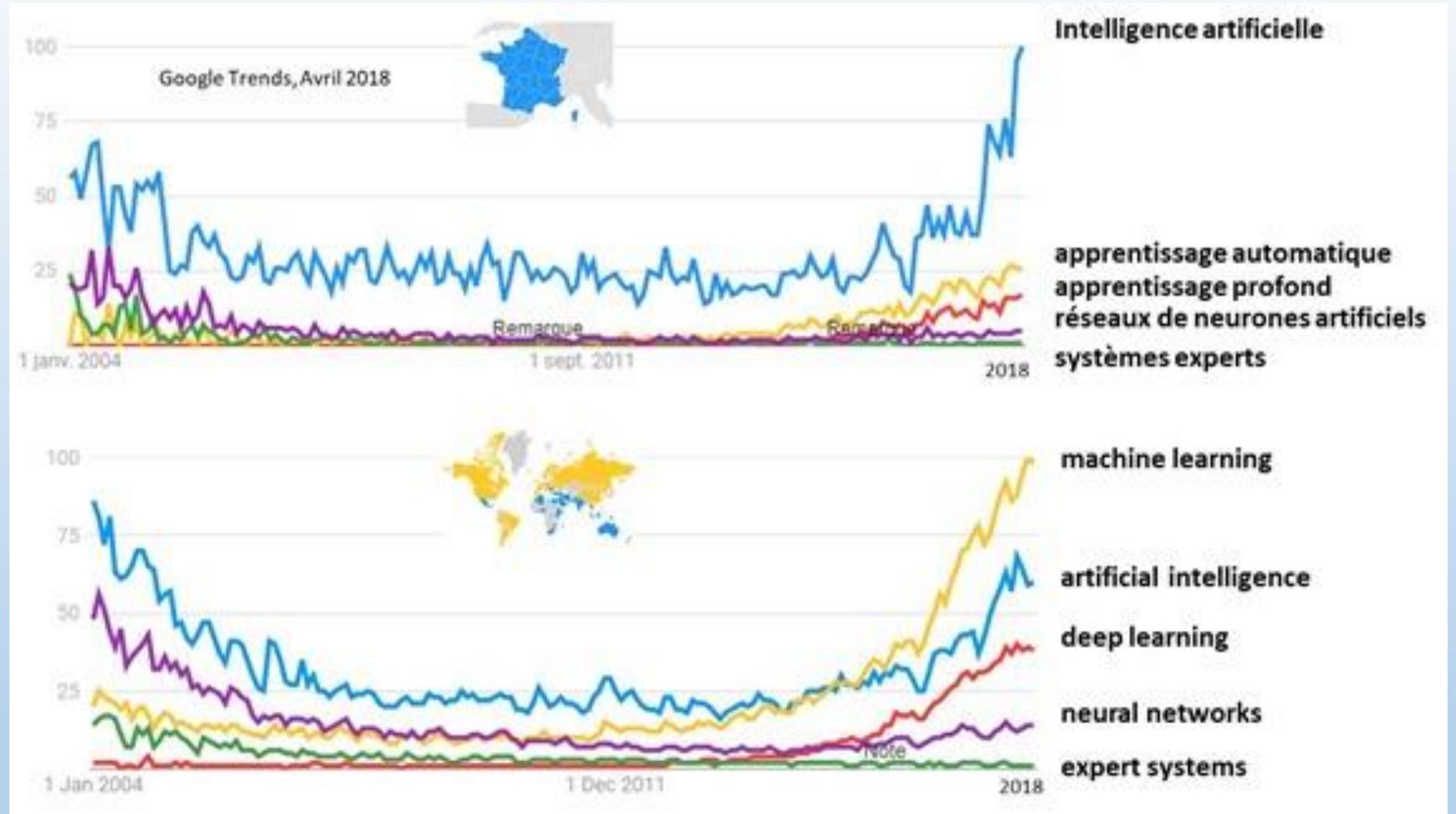
# S.E vs Machine Learning

Système Expert	Machine learning
<p>Basé sur des règles et sur un modèle de faits</p> <p>Nécessite l'identification de toutes les règles : aucune erreur ou omission permise.</p>	<p>Basé sur les données</p> <p>Nécessite la confrontation à des milliers/millions de cas pour renforcer la robustesse des décisions.</p> <p>Requiert la disponibilité d'un large volume de données pertinentes et de bonne qualité.</p>
<p>Algorithmes définis et maîtrisés par l'homme</p> <p>Évolution au gré des concepteurs.</p>	<p>Algorithme engendré par la machine.</p> <p>Évolution continue selon l'apprentissage.</p>
<p>Délai de conception dépendant du type d'entrants : spécialement bien adapté aux tables de décision.</p> <p>Développement rapide</p>	<p>Mise en œuvre longue.</p> <p>L'identification des données utiles et leur contrôle nécessitent une équipe d'experts en science des données (data scientists et data engineers).</p> <p>Investissement dans l'apprentissage.</p>
<p>Grande fiabilité.</p> <p>Résultats stricts, rigoureux (hors maladresse dans l'établissement des règles).</p> <p>Fourni une trace des raisonnements effectués</p>	<p>Risque d'erreur.</p> <p>Risque de décisions erronées (le système est à la mesure des cas appris et peut créer des faux positifs).</p> <p>Raisonnement de type boîte noire : aucune trace</p>



# Conclusions

*Intérêt décroissant du symbolique vs Connexionnisme et Deep Learning*



# Conclusions

## ***Symbolique vs Machine et Deep Learning***

*L'architecture des machines à prédire est transformée pour affronter les big Data*

*La « querelle » entre symbolistes et connexionnistes se poursuit*

*Mais des voix prônent la coopération entre ces deux voies via une hybridation des méthodes et des machines*

*Le rêve pour certains : une IA Généralisée*



# Exemples d'hybridation

La startup franco-américaine Cosmo Tech associe **machine learning**, **symbolisme**, **modélisation de systèmes complexes** avec un langage spécifique (CoSML) et simulation pour créer des **solutions d'entreprises**.

Chez IBM qui est assez **connexionniste** dans l'ensemble, l'approche **cognitiviste** consiste à extraire des règles de textes via du **deep learning** appliqué au traitement du langage puis à les exploiter dans des **systèmes symboliques et de la logique formelle**.



# Conclusions

*Le rôle de **l'intelligence artificielle** et du **système expert** est de fournir un système d'information basé sur la **connaissance** qui devrait avoir des attributs humains afin de **reproduire la capacité humaine** dans la **prise de décision**.*

*La gestion **des connaissances** englobe un large éventail de connaissances transversales :*

- *sciences cognitives,*
- *Bibliothéconomie,*
- *sciences de l'information,*
- *science organisationnelle,*
- *simulations,*
- *base de données rationnelle et objet,*
- *réseaux sémantiques,*
- *systèmes d'aide à la décision,*
- *rédaction technique et communication,*
- *...*

