



HAL
open science

Intelligence Artificielle et Systèmes Multi-Agents

Badr Benmammar

► **To cite this version:**

Badr Benmammar. Intelligence Artificielle et Systèmes Multi-Agents. Engineering school. 2009. cel-00660507

HAL Id: cel-00660507

<https://cel.hal.science/cel-00660507>

Submitted on 16 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Intelligence Artificielle et Systèmes Multi-Agents

Badr Benmammour
bbm@badr-benmammour.com

Plan

❑ La première partie : L'intelligence artificielle (IA)

- ❑ Définition de l'intelligence artificielle (IA)

- ❑ Domaines d'application de l'IA

 - ❑ La reconnaissance de formes

 - ❑ Le traitement automatique des langues

 - ❑ Les systèmes experts

 - ❑ L'apprentissage automatique

- ❑ L'intelligence artificielle distribuée (IAD)

Plan

□ La deuxième partie : La technologie Agent

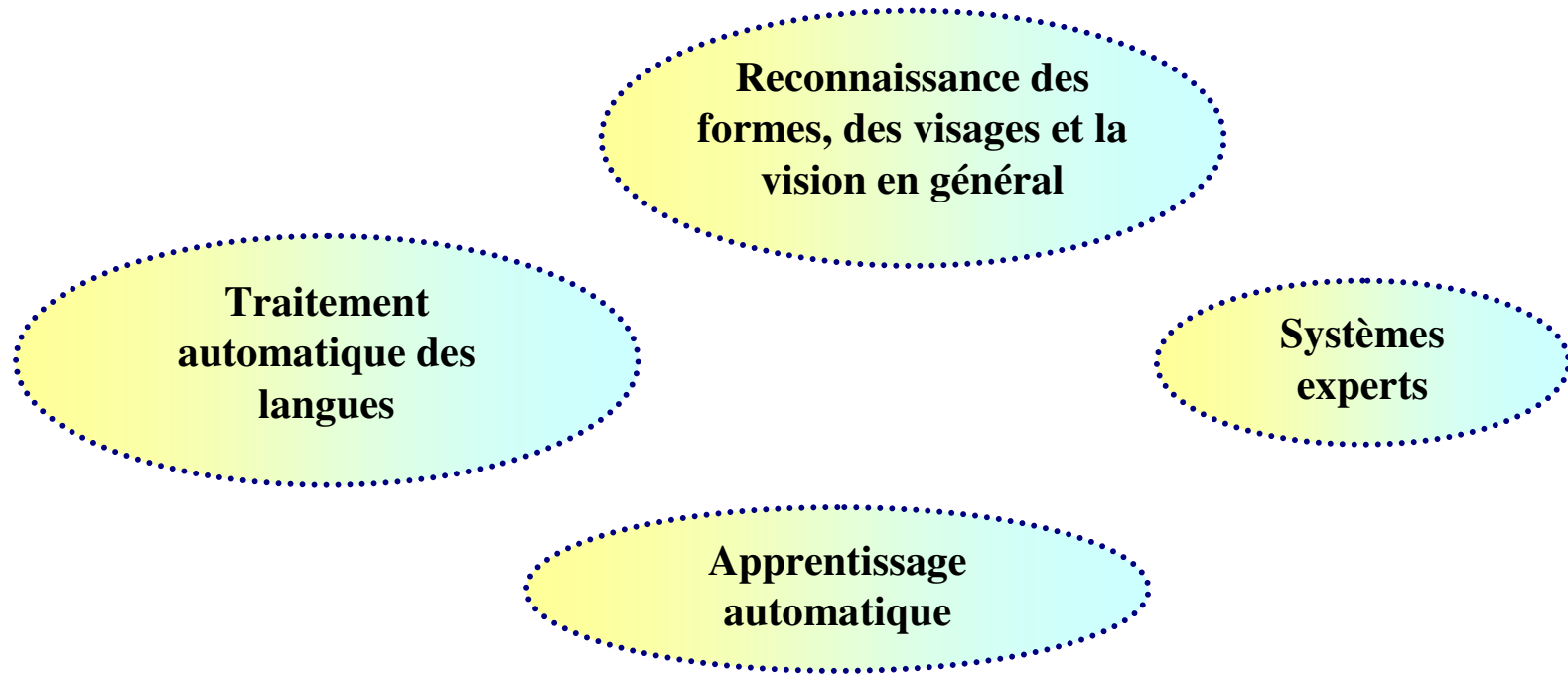
- Définition d'un agent
- Les agents intelligents et les agents mobiles
- Les caractéristiques multidimensionnelles d'un agent
- Architecture concrète pour un agent
 - Les agents logiques
 - Les agents réactifs
 - Les agents BDI
 - Les agents multi-niveaux
- Modèle type d'un agent

La première partie :
L'intelligence artificielle (IA)

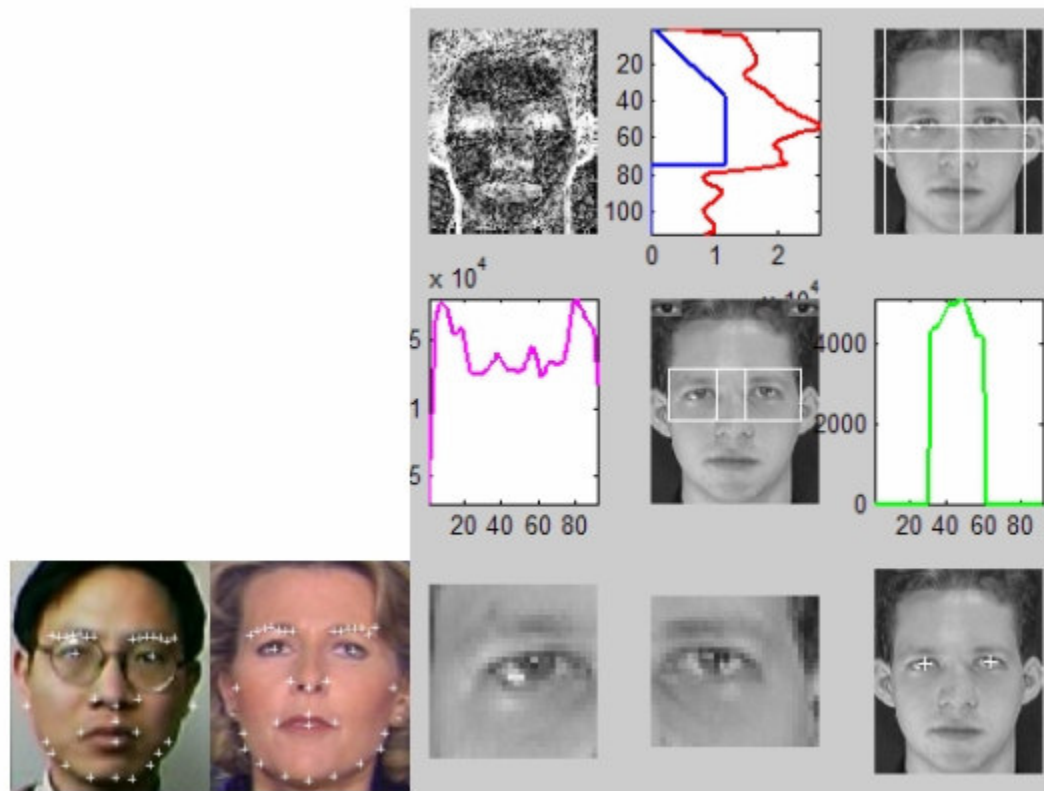
Définition de l'intelligence artificielle (IA)

- ❑ L'IA Désigne la simulation des mécanismes de la pensée par un ordinateur
- ❑ La faculté de reproduire un raisonnement par des moyens informatiques
- ❑ Un ensemble de réalisations et de recherches qui essaient d'imiter artificiellement les performances humaines
 - ❑ L'émergence des systèmes "**adaptatifs**"

Domaines d'application de l'IA



La reconnaissance de formes



La reconnaissance de formes RdF

- **La reconnaissance de formes** (ou parfois **reconnaissance de motifs**) est un ensemble de techniques et méthodes visant à identifier des *motifs* à partir de *données brutes* afin de prendre une décision dépendant de la catégorie attribuée à ce motif.

Définition

- ❑ L'homme est le plus parfait des systèmes de RdF.

- ❑ Reconstitution sur «machine» des fonctions typiquement humaines:
 - ❑ Perception ;

 - ❑ Analyse et représentation ;

 - ❑ Interprétation.

Analogies

Monde Réel



Perception

Données réelles en entrée

(scènes 2D, 3D, parole, ...)

⇒ système de saisie (capteur)

Objet à identifier



Analyse

Transformations appliquées à la forme pour faciliter son interprétation

⇒ extraction de caractéristiques

Représentation



Interprétation

Identification ou catégorisation d'un objet inconnu par rapport à un objet conservé en mémoire par apprentissage

⇒ classification

Classe d'appartenance



Action

Intelligence Artificielle

Systeme de RdF

□ Composantes d'un système de RdF :

- Mécanique (satellite, bras d'un robot, ...)
- Saisie (caméra, scanner, micro, ...)
- Electronique (carte mémoire, CPU, ...)
- ALGORITHMIQUE

Processus de RdF

- ❑ Les informations issues du monde réel sont généralement :
 - ❑ trop volumineuses
 - ❑ peu pertinentes
- ❑ Le processus de RdF est un processus de réduction *progressive* et *sélective* de l'information

Processus de RdF

□ Etapes de traitement d'un processus de RdF:

+

□ Prétraitements

binarisation, segmentation, élimination du bruit, normalisation, ...

□ Extraction des informations pertinentes

parole: fréquence, ...

image: contours, ...

□ Représentation de ces informations en vue de leur classification

vecteur, graphe,

□ Classification de la forme

apprentissage/décision,

Quantité
d'informations

-

Le traitement automatique des langues



Le traitement automatique des langues

- ❑ **Le Traitement automatique des langues** est une discipline à la frontière de :
 - ❑ La linguistique ;
 - ❑ L'informatique ;
 - ❑ L'intelligence artificielle.
- ❑ Concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain.

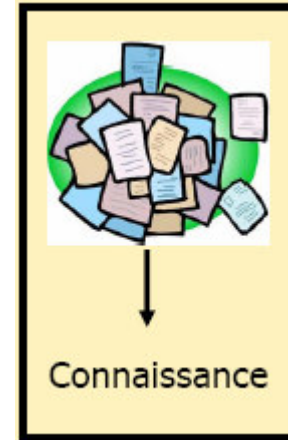
Applications

- ❑ La traduction automatique ;
- ❑ La correction orthographique ;
- ❑ La recherche d'information et la fouille de textes ;
- ❑ Le résumé automatique de texte ;
- ❑ La génération automatique de textes ;

Fouille de Texte (Text Mining)

Qu'est-ce que le Text Mining ?

- ❑ Le Text Mining est l'ensemble des :
 - ❑ Technologies et méthodes
 - ❑ ... destinées au traitement automatique
 - ❑ ... de données textuelles
 - ❑ ... disponibles sous forme informatique,
 - ❑ ... en assez grande quantité
 - ❑ ... en vue d'en dégager et structurer le contenu, les thèmes dans une perspective d'analyse rapide de **découverte d'informations cachées ou de prise automatique de décision**

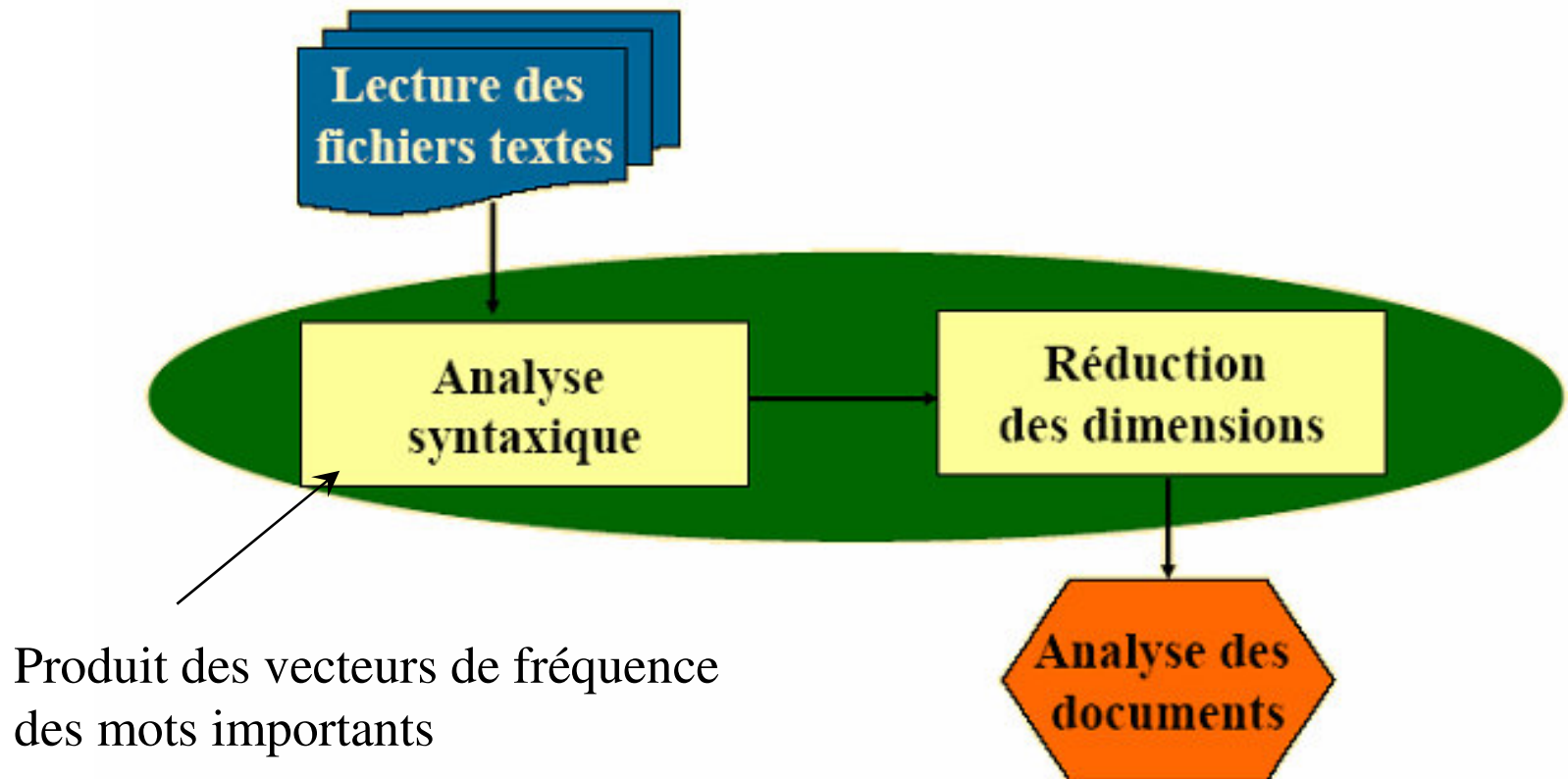


Définition

- Text Mining

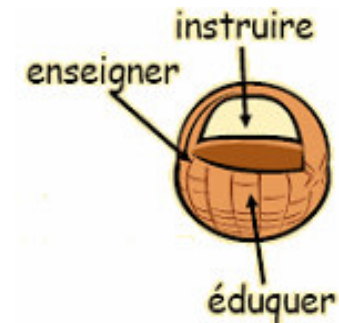
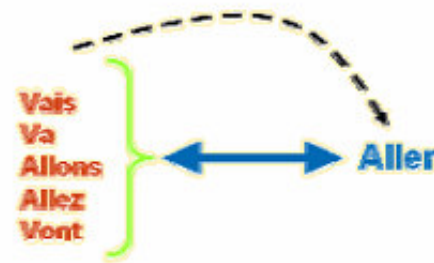
- Procédé consistant à synthétiser (classer, structurer, résumer, ...) les textes en analysant les relations et les règles entre unités textuelles (mots, groupes, phrases, documents)

Processus de Text Mining: Vue simplifiée



Analyse et Préparation

- ❑ Corriger l'orthographe
- ❑ Eliminer les mots vides
- ❑ Découper les textes en unités
- ❑ Associer des termes à une catégorie grammaticale ou sémantique
- ❑ Réduire le nombre de termes à traiter



Calculs de fréquence des termes

Table de fréquences d'apparition des termes

| | Mot 1 | Mot 2 | Mot 3 | Mot 4 |
|-------|-------|-------|-------|-------|
| Doc 1 | 1 | 2 | 0 | 3 |
| Doc 2 | 5 | 1 | 1 | 5 |
| Doc 3 | 2 | 0 | 0 | 2 |

Réduction des dimensions

- ❑ Réduire les dimensions de la table de fréquences
 - ❑ En déterminant les termes les plus significatifs
 - ❑ En groupant les termes par affinité (profile)

Domaines d'application

❑ Exploration du contenu des documents

- ❑ Questions ouvertes dans une enquête
- ❑ Commentaires et plaintes des clients
- ❑ Analyse des réclamations de garantie

❑ Affectation de documents à des thèmes prédéfinis

- ❑ Traitement des e-mails (redirection, filtrage)

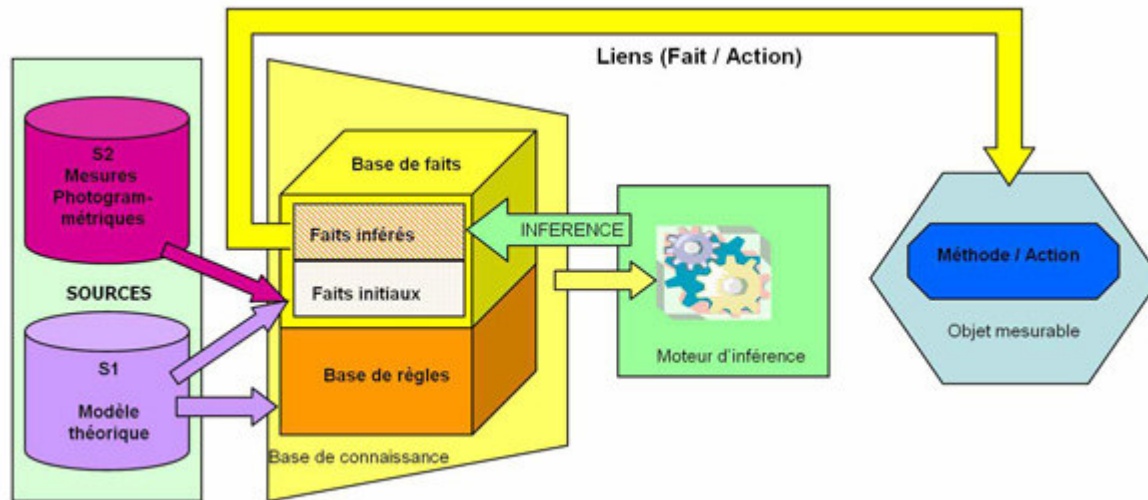
❑ Recherche d'information

- ❑ Interrogation de textes par concepts, mots-clés, sujets, phrases visant à obtenir des résultats triés par ordre de pertinence, à la Google

Étapes de la fouille de textes

- ❑ Sélection de textes
- ❑ Extraction des termes
 - ❑ Analyse grammaticale
 - ❑ Filtrage des termes extraits
- ❑ Transformation
 - ❑ Réduction des dimensions
- ❑ Classification
- ❑ Visualisation des résultats
- ❑ Interprétation des résultats

Les systèmes experts



Les systèmes experts

- Un système expert est un logiciel capable de répondre à des questions, en effectuant un raisonnement à partir de **faits et de règles** connus. Il peut servir notamment comme outil d'aide à la décision.

Les systèmes experts

- Un système expert se compose de 3 parties :
 - Une base de faits ;
 - Une base de règles ;
 - Un moteur d'inférence.

Faits et règles

- ❑ Faits

- ❑ Ensemble de vérités connues

- ❑ Règles

- ❑ Partie condition : prémisses
 - ❑ Partie action : conclusion, calcul, affichage ...
 - ❑ Éventuellement : coefficient de confiance

SI *condition* ALORS *action* [*coefficient*]

Les systèmes experts

- Pour l'essentiel, ils utilisent la règle d'inférence suivante :
- Si P est vrai (*fait* ou *prémisse*) et si on sait que P implique Q (*règle*) alors, Q est vrai (*nouveau fait* ou *conclusion*).

Le moteur d'inférence

Le moteur d'inférence est capable d'utiliser faits et règles pour produire de nouveaux faits, jusqu'à parvenir à la réponse à la question experte posée.

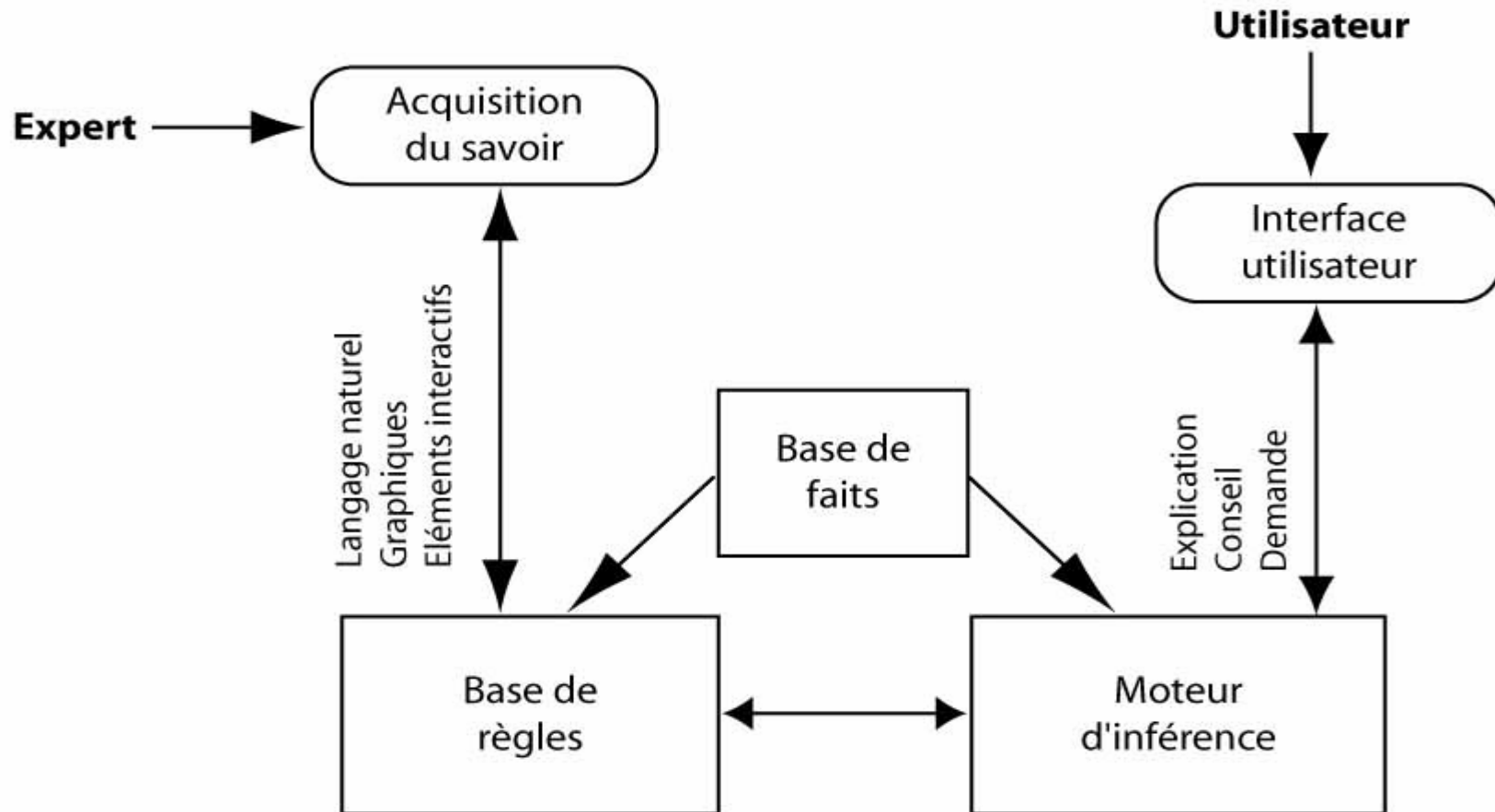
La plupart des systèmes experts existants reposent sur des mécanismes de logique formelle et utilisent le raisonnement déductif.

Pourquoi un SE ?

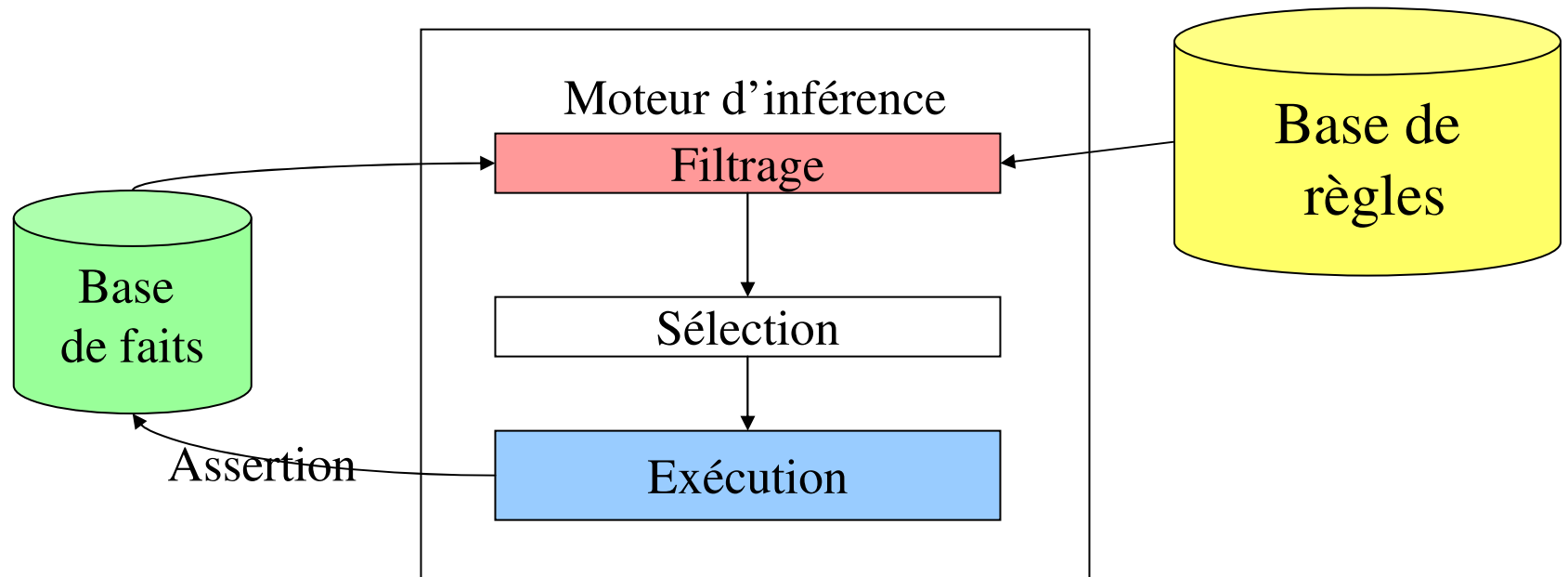
- ❑ Problèmes mal formalisés ou dont la formulation évolue
 - ❑ Médecine : nouveaux examens, nouvelles connaissances, nouvelles thérapies

- ❑ Pas d'algorithme connu (ou algorithme trop complexe)
 - ❑ Reconnaissance de l'écriture
 - ❑ Jeu d'échecs
 - ❑ Affectation des ressources, ordonnancement

Les systèmes experts



Architecture



Mode de raisonnements

- ❑ Il existe de nombreux types de moteurs, capables de traiter différentes formes de règles logiques pour déduire de nouveaux faits à partir de la base de connaissance.
- ❑ On distingue souvent trois catégories, basées sur la manière dont les problèmes sont résolus :
 - ❑ Les moteurs - dit à « *chaînage avant* » - qui partent des faits et règles de la base de connaissance, et tentent de s'approcher des faits recherchés par le problème.
 - ❑ Les moteurs - dits à « *chaînage arrière* » - qui partent des faits recherchés par le problème, et tentent par l'intermédiaire des règles, de « remonter » à des faits connus,
 - ❑ Les moteurs - dits à « *chaînage mixte* » - qui utilisent une combinaison de ces deux approches *chaînage avant* et *chaînage arrière*.

Chaînage avant

Raisonnement guidé par les données :

- Détecter les règles dont les prémisses sont vérifiées (filtrage)
- Sélectionner la règle à appliquer
- Appliquer la règle
- Recommencer jusqu'à ce qu'il n'y ait plus de règle applicable

Chaînage arrière

- ❑ But initial placé au sommet d'une pile
- ❑ Détection des règles qui concluent à ce but
- ❑ Résolution de conflits
- ❑ Application de la règles, i.e, les éléments des prémisses deviennent de nouveau sous- buts à atteindre.
- ❑ Arrêt : pile vide ou aucune règle applicable

Exercice

❑ Exercice 1

❑ Soit la base de règles suivantes :

❑ R1 : $(\neg A \rightarrow B) \rightarrow P$

❑ R2 : $(P \text{ et } Q) \rightarrow F$

❑ R3 : $(C \rightarrow A) \rightarrow Q$

❑ R4 : $F \rightarrow (D \rightarrow K)$

❑ R5 : $K \rightarrow (M \text{ et } L)$

❑ La base initiale de faits est : (A, D).

❑ Prouvez le fait M par chaînage avant.

Exercice

❑ Exercice 2

❑ Un expert a construit la base de règles suivantes :

❑ R1 : A et B \rightarrow C

❑ R2 : D \rightarrow A

❑ R3 : E \rightarrow F

❑ R4 : G \rightarrow H

❑ R5 : I \rightarrow F

❑ R6 : H et F et J \rightarrow B

❑ R7 : H et K \rightarrow J

❑ R8 : G et F \rightarrow K

❑ La base initiale de faits est : (D, G, I).

❑ Prouvez le fait C par chaînage avant et chaînage arrière.

Exercice

□ Exercice 3

□ Soit la base de règles suivantes :

□ R1: $(A \text{ et } E) \rightarrow G$

□ R2: $(E \rightarrow K) \rightarrow D$

□ R3: $(A \text{ et } B) \rightarrow (C \text{ et } D)$

□ R4: $(B \rightarrow A) \rightarrow ((K \text{ et } \neg F) \text{ ou } G)$

□ R5: $(A \text{ et } G) \rightarrow B$

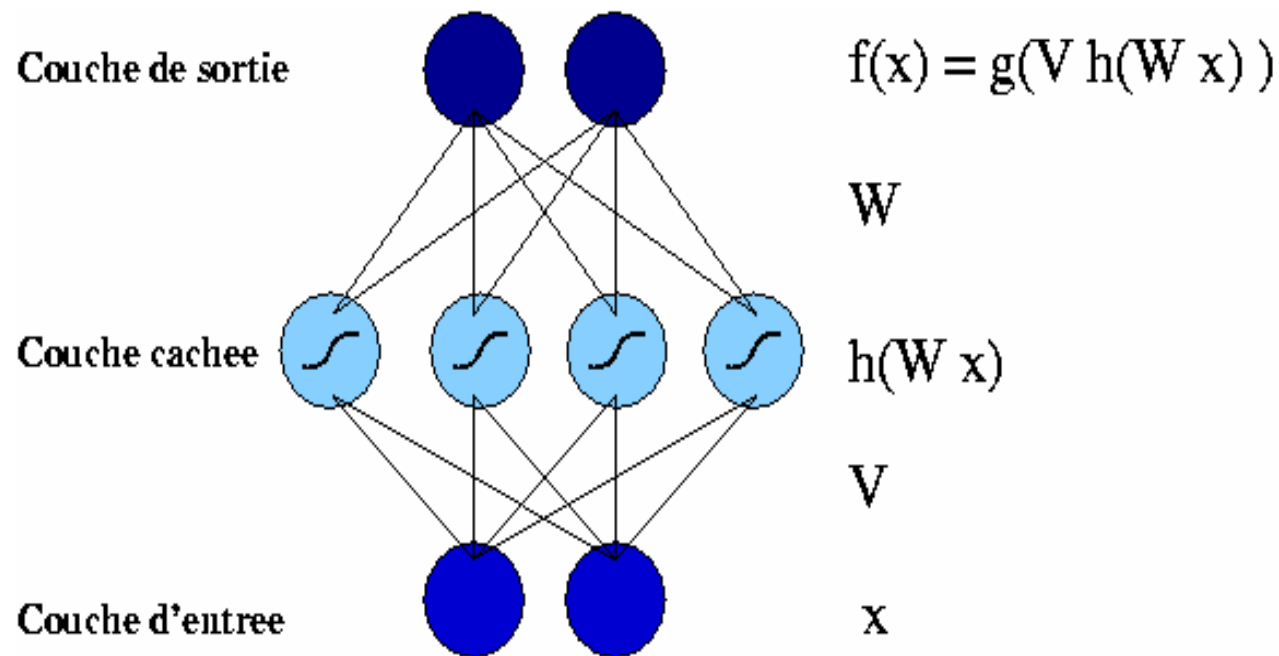
□ R6: $(K \rightarrow E) \rightarrow C$

□ R7: $(F \text{ et } E) \rightarrow B$

□ La base initiale de faits est : (A, F) .

□ Prouvez le fait C par chaînage avant.

L'apprentissage automatique



L'apprentissage automatique

- ❑ L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à **un processus d'apprentissage**, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.
- ❑ **Objectif** : extraire et exploiter automatiquement l'information présente dans un jeu de données.

Types d'apprentissage

□ Les algorithmes d'apprentissage peuvent se catégoriser selon le type d'apprentissage qu'ils emploient :

□ L'apprentissage supervisé

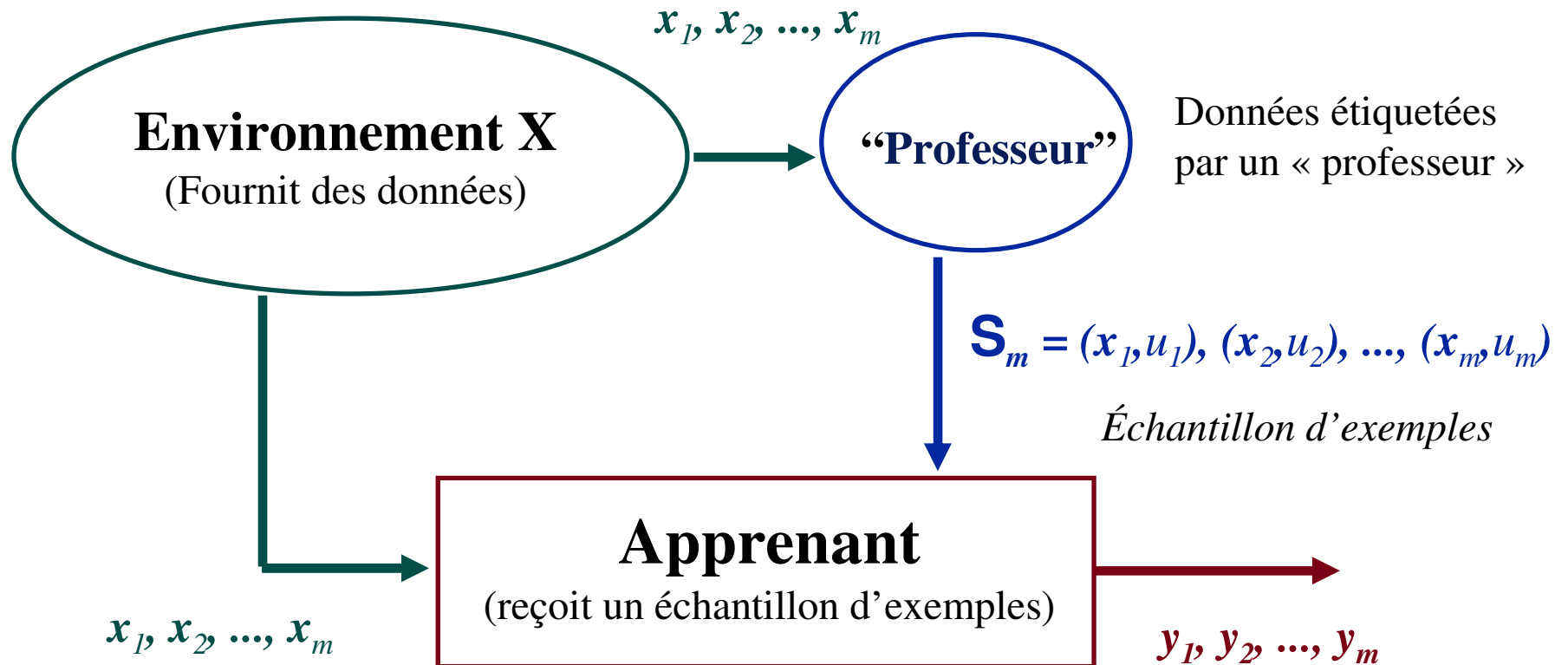
□ L'apprentissage non-supervisé

□ L'apprentissage par renforcement

L'apprentissage supervisé

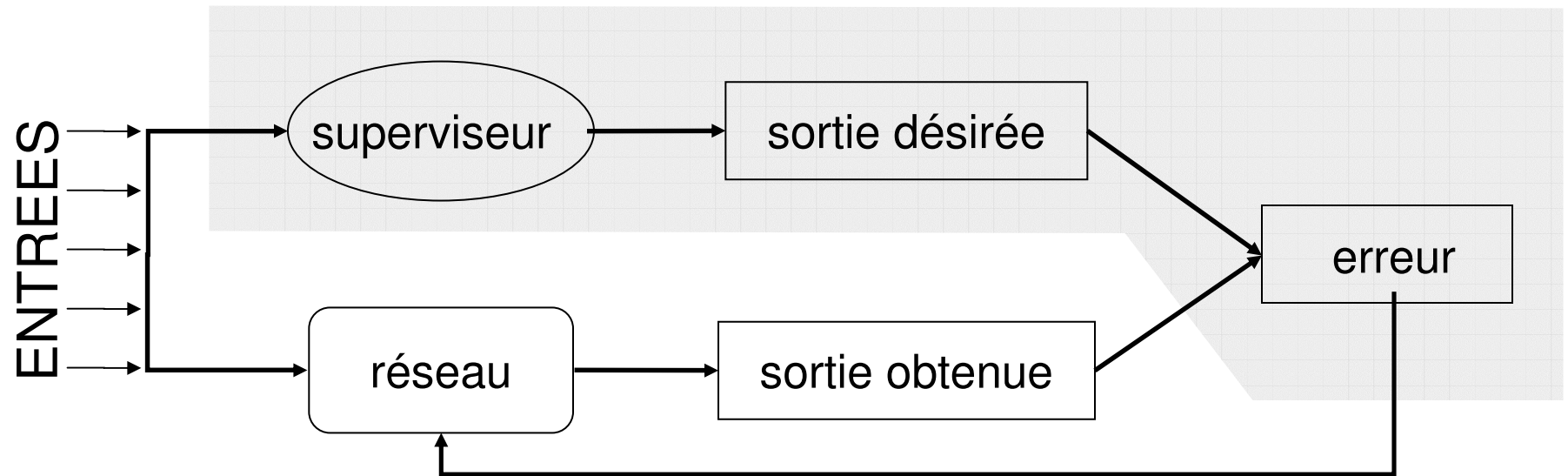
- Un expert est employé pour étiqueter correctement des exemples.
 - L'apprenant doit alors trouver ou approximer la fonction qui permet d'affecter la bonne étiquette à ces exemples.

L'apprentissage supervisé



Approximer au mieux la sortie désirée pour chaque entrée observée

Apprentissage supervisé



L'apprentissage non supervisé

- ❑ Aucun expert n'est disponible.
 - ❑ L'algorithme doit découvrir par lui-même la structure des données.
 - ❑ Le clustering est un algorithme d'apprentissage non supervisés.

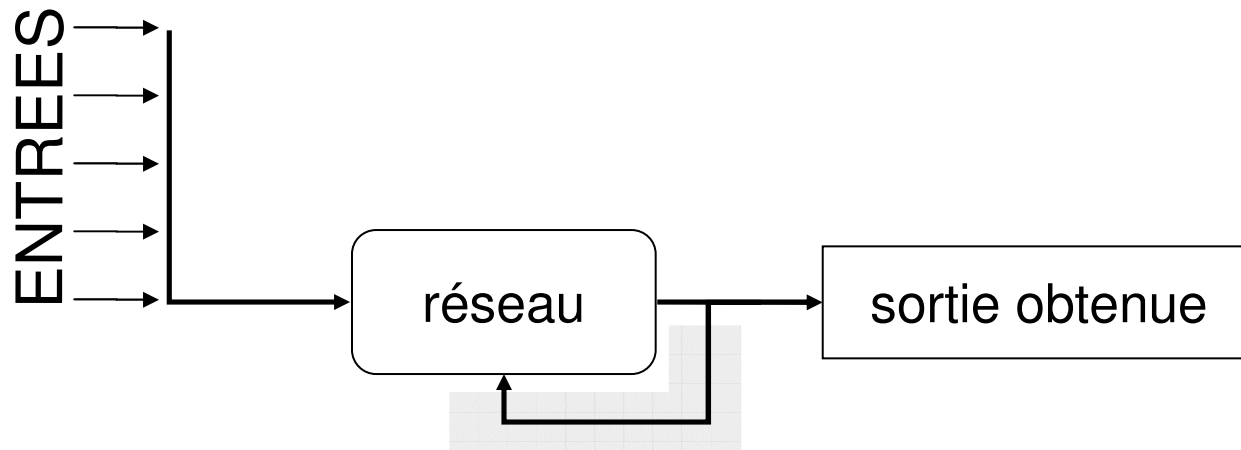
L'apprentissage non supervisé

A partir de l'échantillon d'apprentissage $S = \{(x_i)\}_{1,m}$
non étiqueté,
on cherche des **régularités** sous-jacente

- Sous forme d'une fonction
- Sous forme d'un modèle complexe

afin de résumer, détecter des régularités, comprendre...

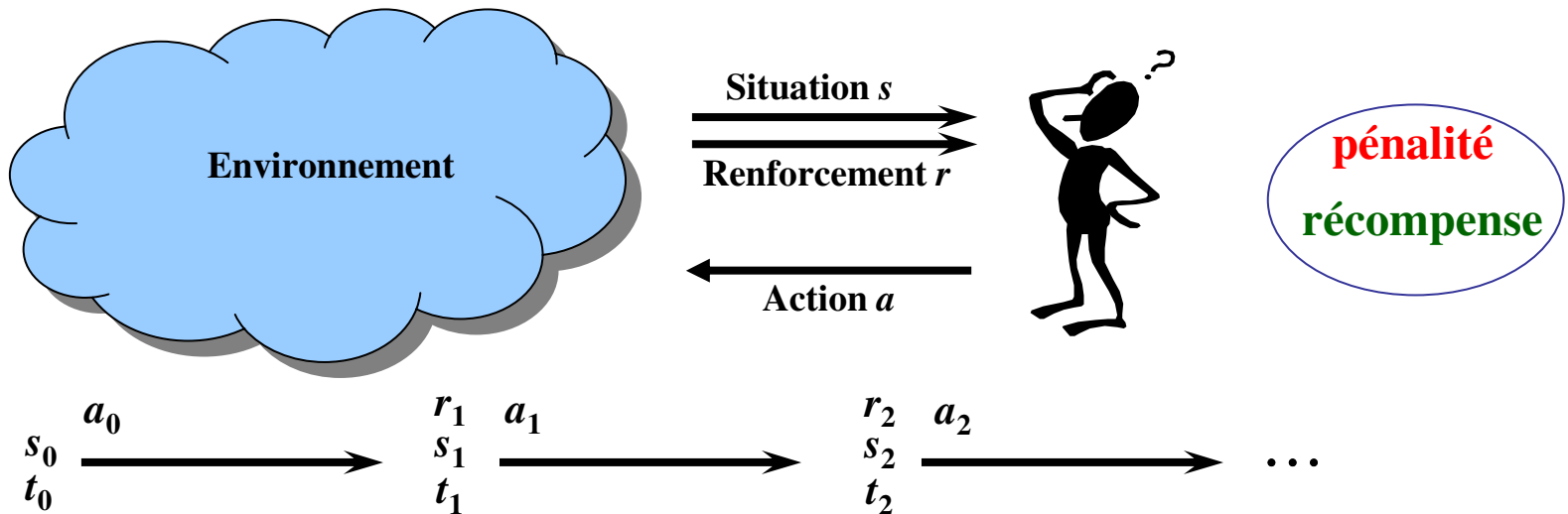
Apprentissage non supervisé



Apprentissage par renforcement (AR)

- ❑ L'algorithme apprend un comportement étant donné une observation.
 - ❑ L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage.

Apprentissage par renforcement (AR)



- ❑ L'agent apprend à se rapprocher d'une **stratégie comportementale optimale** par des interactions répétitives avec l'environnement
- ❑ Les décisions sont prises séquentiellement à des intervalles de temps discrets

Pourquoi l'AR ?

- Il est très utile dans le cadre de problèmes où :
 - des stratégies comportementales efficaces sont inconnues a priori ou sont difficilement automatisables
 - lorsqu'il y a de l'incertain dans la manière dont l'environnement évolue
- L'AR se distingue des autres approches d'apprentissage par plusieurs aspects :
 - L'apprentissage se fait sans supervision
 - Il repose sur le principe d'essai/erreur

Limitations du cadre de l'AR

- ❑ Quand les espaces d'états et d'actions sont très grands
 - ❑ Mémoire nécessaire pour représenter une table
 - ❑ Temps nécessaire pour pouvoir remplir efficacement cette table

Applications

Les jeux

- Backgammon

- Échecs

Applications industrielles

- Routage de paquets

- Contrôle et maintenance de machines

- Contrôle d'une flotte d'ascenseurs

Quelques algorithmes

- La méthode des k plus proches voisins**
- Les réseaux de neurones**
- Les arbres de décision**
- Les algorithmes génétiques**

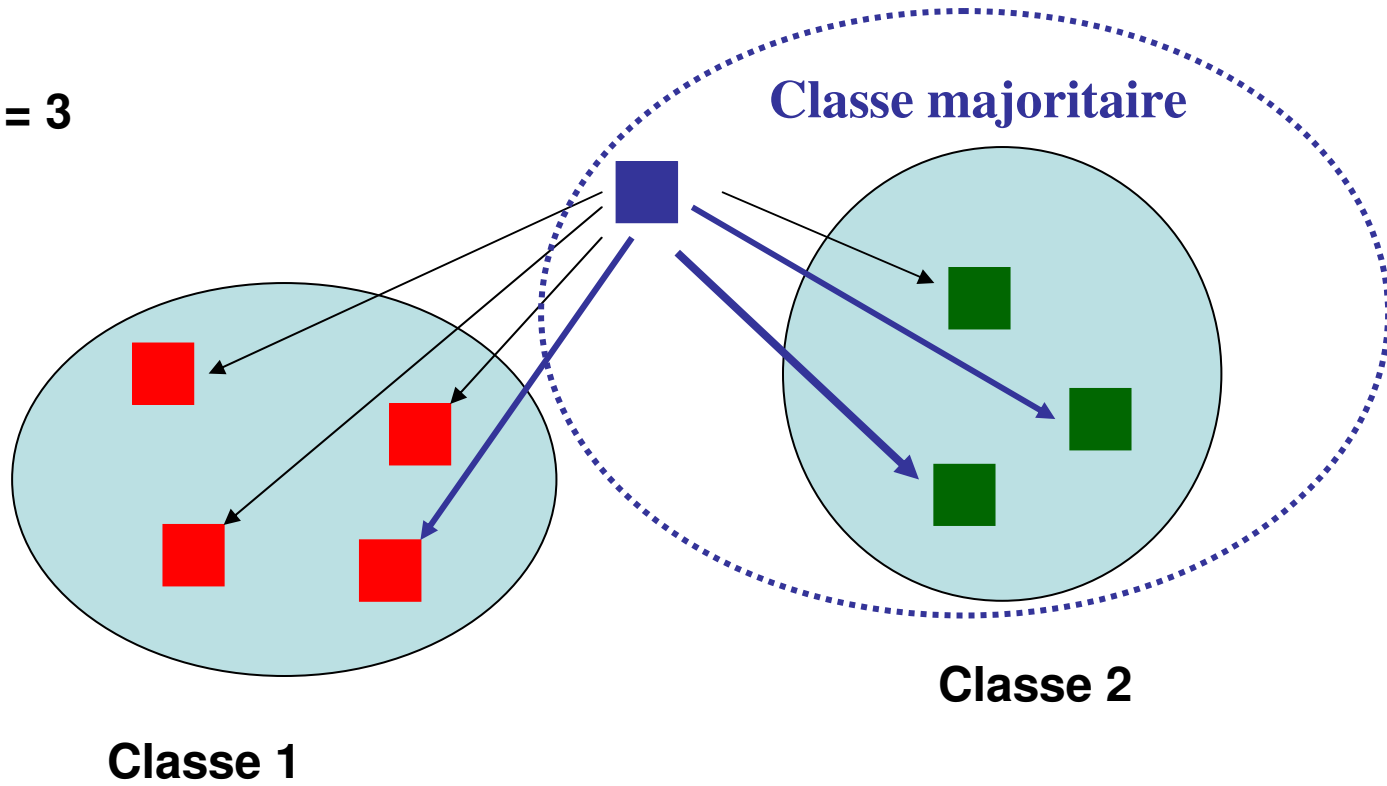
Méthode des k plus proches voisins

Algorithme

- ❑ On dispose d'une base de données d'apprentissage constituée de m couples « entrée-sortie ».
- ❑ Pour estimer la sortie associée à une nouvelle entrée x , la méthode consiste à prendre en compte les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x , selon une distance à définir.
- ❑ Par exemple dans un problème de **classification**, on retiendra **la classe la plus représentée** parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x .

Exemple

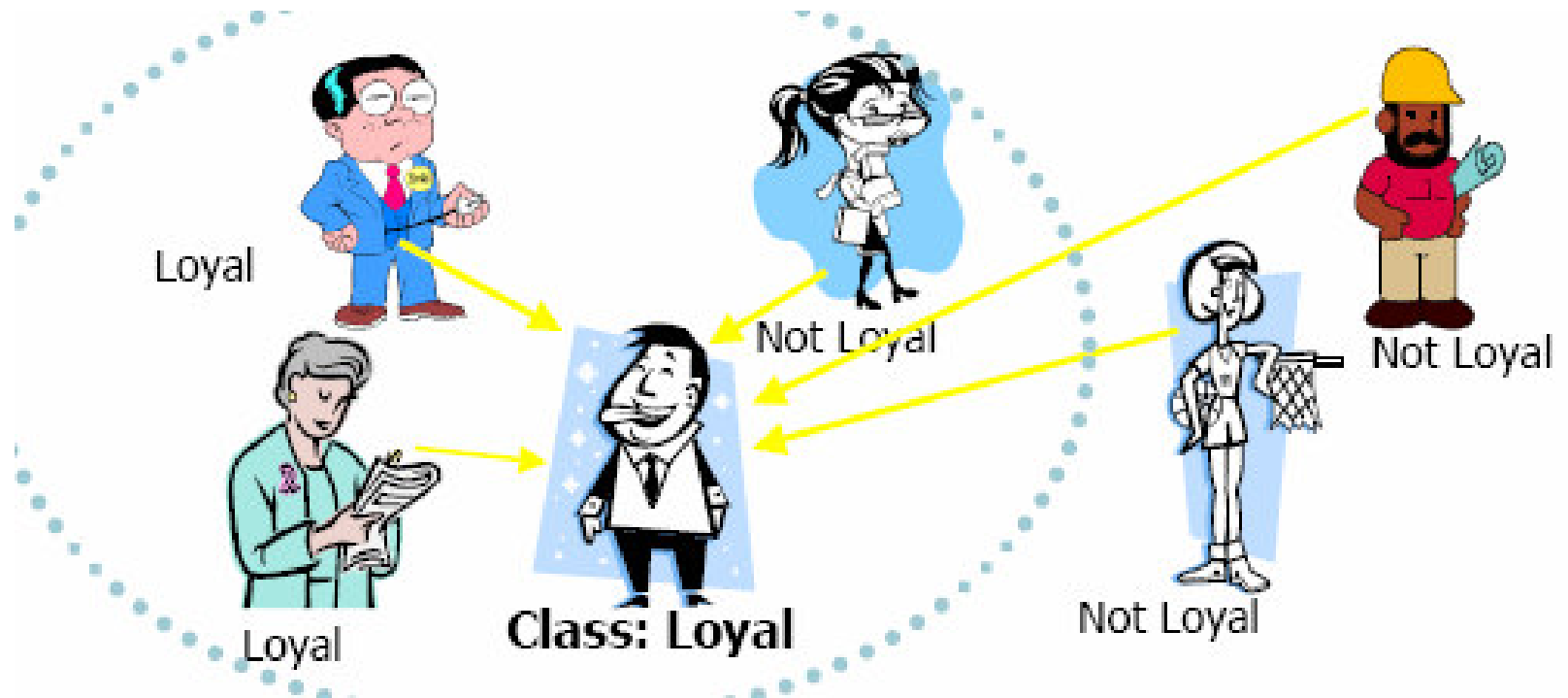
K = 3









Algorithme

- ❑ **Paramètre** : le nombre k de voisins
- ❑ **Donnée** : un échantillon de m exemples et leurs classes
 - ❑ La classe d'un exemple X est $c(X)$
- ❑ **Entrée** : un enregistrement Y
- ❑ Déterminer les k plus proches exemples de Y en calculant les distances
- ❑ Combiner les classes de ces k exemples en une classe c
- ❑ **Sortie** : la classe de Y est $c(Y)=c$

Exemple: Client loyal ou non









Exemple

| Customer | Age | Income | No. credit cards | Loyal |
|--|-----|--------|------------------|-------|
| John  | 35 | 35K | 3 | No |
| Rachel  | 22 | 50K | 2 | Yes |
| Hannah  | 63 | 200K | 1 | No |
| Tom  | 59 | 170K | 1 | No |
| Nellie  | 25 | 40K | 4 | Yes |
| David  | 37 | 50K | 2 | ? |

Exemple

K=3

| Customer | Age | Income | No. credit cards | Loyal |
|--|-----|--------|------------------|-------|
| John  | 35 | 35K | 3 | No |
| Rachel  | 22 | 50K | 2 | Yes |
| Hannah  | 63 | 200K | 1 | No |
| Tom  | 59 | 170K | 1 | No |
| Nellie  | 25 | 40K | 4 | Yes |
| David  | 37 | 50K | 2 | Yes |

| Distance from David |
|---|
| $\text{sqrt} [(35-37)^2+(35-50)^2+(3-2)^2]=15.16$ |
| $\text{sqrt} [(22-37)^2+(50-50)^2+(2-2)^2]=15$ |
| $\text{sqrt} [(63-37)^2+(200-50)^2+(1-2)^2]=152.23$ |
| $\text{sqrt} [(59-37)^2+(170-50)^2+(1-2)^2]=122$ |
| $\text{sqrt} [(25-37)^2+(40-50)^2+(4-2)^2]=15.74$ |

Classe majoritaire

Distance

- ❑ Le choix de la distance est primordial au bon fonctionnement de la méthode
- ❑ Les distances les plus simples permettent d'obtenir des résultats satisfaisants
- ❑ Propriétés de la distance:
 - ❑ $d(A,A)=0$
 - ❑ $d(A,B)=d(B,A)$
 - ❑ $d(A,B)\leq d(A,C) + d(B,C)$

Distance

- $d(x,y) = |x-y|$

ou

- $d(x,y) = |x-y|/d_{\max}$, où d_{\max} est la distance maximale entre deux

numériques du domaine considéré

Distance

- Données binaires : 0 ou 1.

 - On choisit $d(0,0)=d(1,1)=0$ et $d(0,1)=d(1,0)=1$.

- Données énumératives :

 - La distance vaut 0 si les valeurs sont égales et 1 sinon.

- Données énumératives ordonnées : elles peuvent être considérées comme des valeurs énumératives mais on peut également définir une distance utilisant la relation d'ordre.

 - Exemple: Si un champ prend les valeurs A, B, C, D et E, on peut définir la distance en considérant 5 points de l'intervalle $[0,1]$ avec une distance de 0,25 entre deux points successifs, on a alors $d(A,B)=0,25$; $d(A,C)=0,5$; ...

Distance Euclidienne

- Soit $X = (x_1, \dots, x_n)$ et $Y = (y_1, \dots, y_n)$ deux exemples, la distance euclidienne entre X et Y est:

$$D(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Autres distances

Sommation:

$$D(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Distance euclidienne pondérée:

$$D(X, Y) = \sqrt{\sum_{i=1}^n w_i (x_i - y_i)^2}$$

Pourquoi pondérer les attributs?

- ❑ Certains attributs peuvent dominer le calcul de la distance
- ❑ Exemple:

$$\text{Distance (John, Rachel)} = \sqrt{[(35-22)^2 + (35,000-50,000)^2 + (3-2)^2]}$$

Mise en oeuvre de la méthode

- ❑ Choisir les attributs pertinents pour la tâche de classification.
- ❑ Choix de la distance par champ et du mode de combinaison des distances en fonction du type des champs et des connaissances préalables du problème.
- ❑ Choix du nombre k de voisins.
 - ❑ Une heuristique fréquemment utilisée est de prendre k égal au nombre d'attributs plus 1.

Discussion

- ❑ La méthode peut s'appliquer dès qu'il est possible de définir une distance sur les champs
- ❑ La méthode permet de traiter des problèmes avec un grand nombre d'attributs.
 - ❑ Plus le nombre d'attributs est important, plus le nombre d'exemples doit être grand.
- ❑ Les performances de la méthode dépendent du choix de la distance et du nombre de voisins.

Exercice

- ❑ On dispose d'une base de données d'apprentissage constituée de 5 couples « entrée-sortie ».
 - ❑ (Dupont, Admis), (Fernand, Admis), (David, Ajourné), (Dumont, Ajourné) et (Billaut, Admis).
- ❑ Pour chaque étudiant, on dispose aussi de 4 notes dans 4 matières différentes.
 - ❑ Dupont → 14, 12, 8 et 12.
 - ❑ Fernand → 12, 12, 6 et 10.
 - ❑ David → 8, 9, 9 et 1.
 - ❑ Dumont → 15, 11, 3 et 5.
 - ❑ Billaut → 12, 9, 14 et 11.
- ❑ On dispose maintenant d'une nouvelle entrée Verneuil qui a pour notes : 9, 14, 15 et 6.
- ❑ En utilisant la méthode des k plus proches voisins ($k=3$) et en choisissant la distance suivante : $\text{Distance} = \text{Racine}(\text{Somme}(\text{ABS}(X_i - X_j)))$. Déterminez la classe de Verneuil.

Réseaux de Neurones Artificiels

Définition

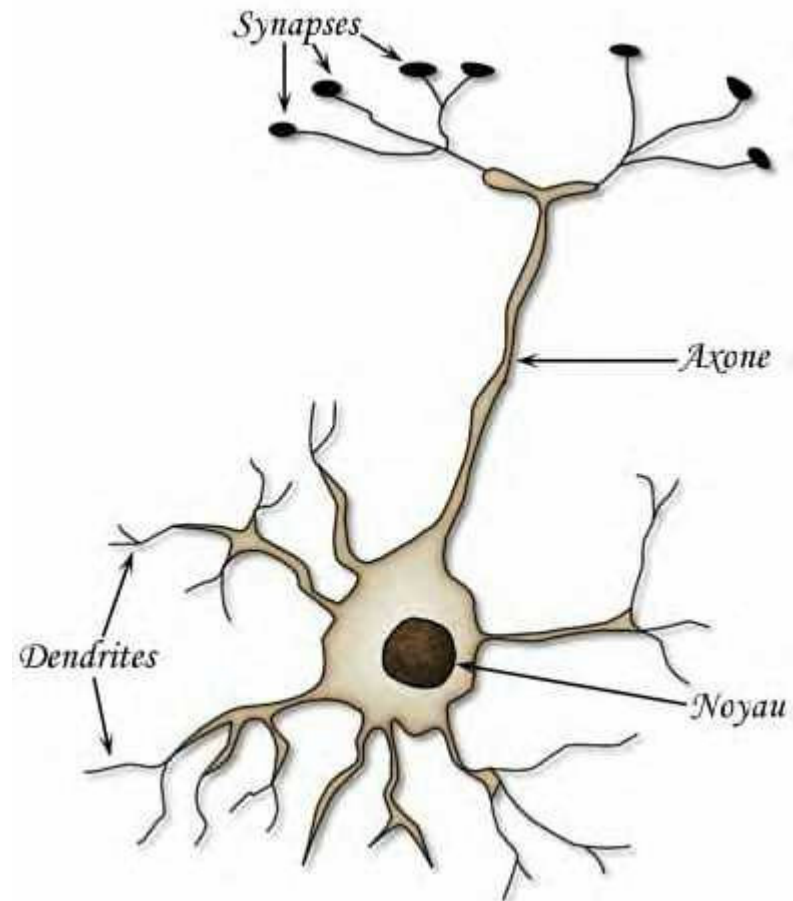
- ❑ Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle.
- ❑ Chaque processeur élémentaire (neurone artificiel) calcule une sortie unique sur la base des informations qu'il reçoit.
- ❑ Inspiration naturelle : analogie avec le cerveau

Fondements Biologiques

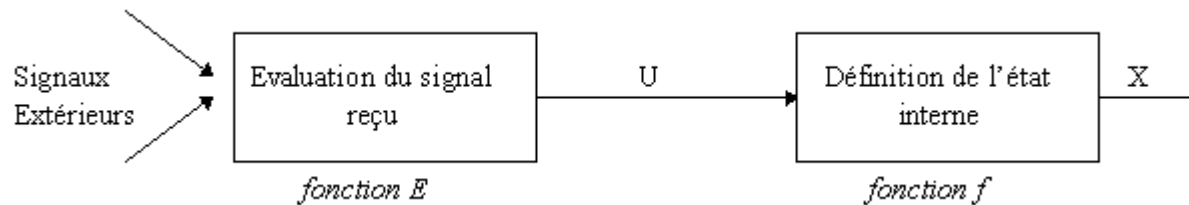
Fonctionnement des neurones

L'influx nerveux est assimilable à un signal électrique se propageant comme ceci :

- ❑ Les dendrites reçoivent l'influx nerveux d'autres neurones.
- ❑ Le neurone évalue l'ensemble de la stimulation reçue.
- ❑ Si elle est suffisante, il est excité : il transmet un signal (0/1) le long de l'axone.
- ❑ L'excitation est propagée jusqu'aux autres neurones qui y sont connectés via les synapses.



Le Neurone Formel



Le neurone formel, l'unité élémentaire d'un RNA, se compose de deux parties :

- ❑ évaluation de la stimulation reçue (fonction E)

- ❑ $E(\text{entrées}) = U$

- ❑ évaluation de son activation (fonction f)

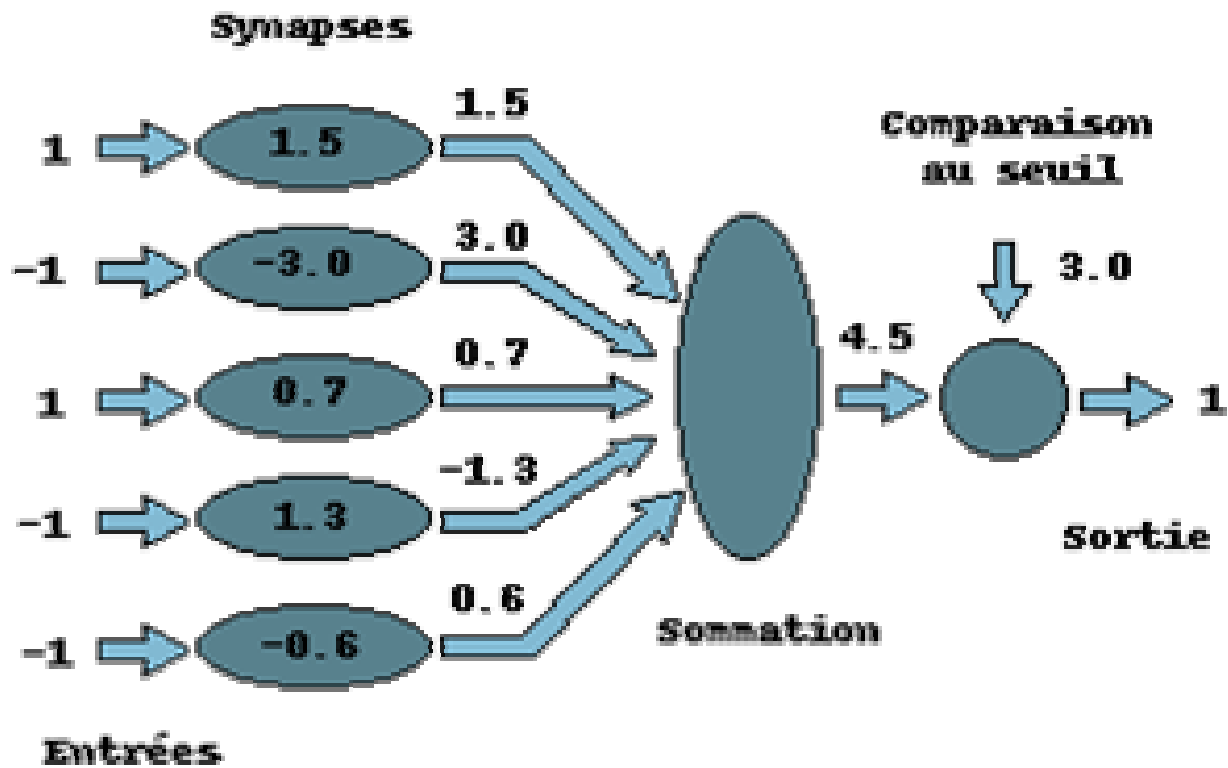
- ❑ $f(U) = X$

Le Neurone Formel

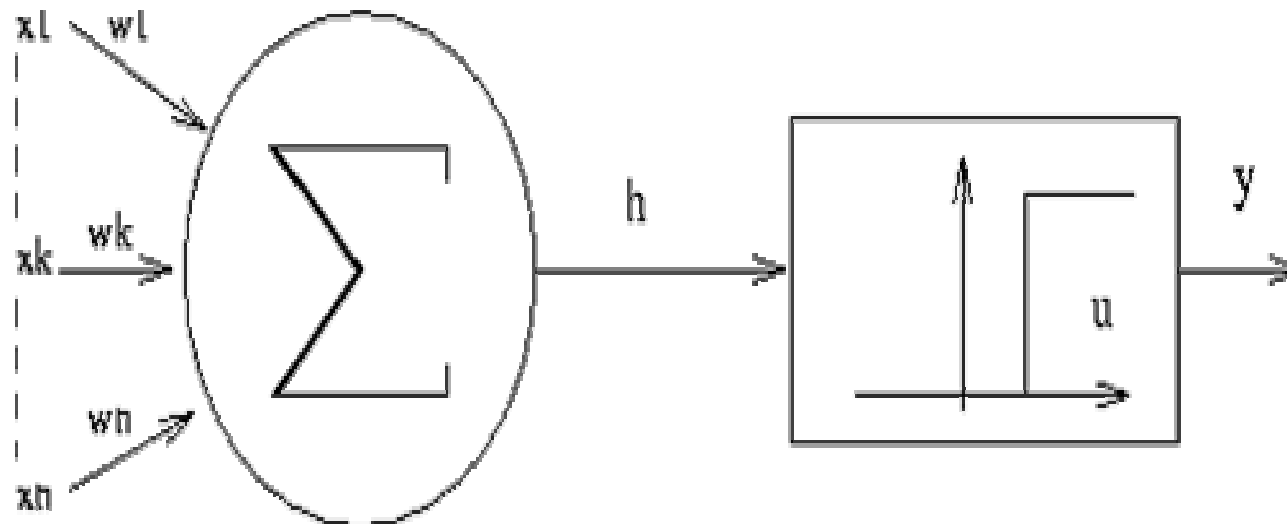
La fonction d'entrée : somme pondérée des signaux d'entrée

$$U_i = E(x_1, \dots, x_j, \dots, x_n) = \sum_{j=1}^n W_{ij} x_j$$

Le Neurone de McCulloch & Pitts



Interprétation mathématique



$$y = f\left(\sum_{j=1}^n w_j x_j - \theta\right)$$

Domaines d'application

□ Classification

- Répartir en plusieurs classes des objets
- Données quantitatives → informations qualitatives
- Reconnaissance des formes

□ Recherche Opérationnelle

- Résoudre des problèmes dont on ne connaît pas la solution

L'intelligence artificielle distribuée (IAD)

- ❑ La branche de l'IA qui s'intéresse à la modélisation du comportement « intelligent » par la coopération entre un ensemble d'agents (distribuer l'intelligence sur plusieurs entités).
- ❑ L'intelligence artificielle classique, modélise le comportement intelligent d'un seul agent.
- ❑ Passage du comportement individuel aux comportements collectifs pour combler les limites de l'IA classique à résoudre des problèmes complexes.

La technologie Agent

Définition d'un agent

- Un agent est une **entité physique ou virtuelle** qui agit dans un environnement, communique directement avec d'autres agents, possède des ressources propres, est capable de percevoir partiellement son environnement et possède des compétences.
- En fonction des ressources, des compétences et des communications, un agent tend à satisfaire ses objectifs.

Définition d'un agent

- ❑ Un agent est **un système informatique**, situé dans un environnement, qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

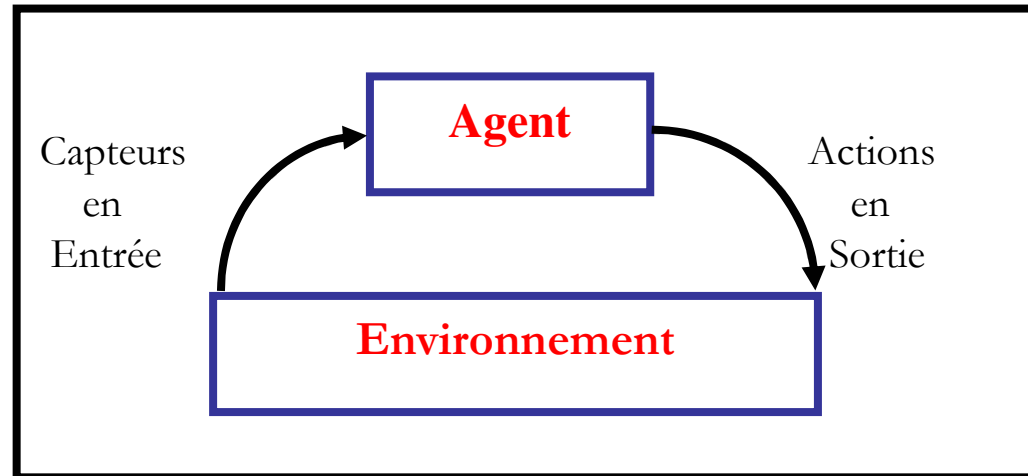
- ❑ Un agent est **un morceau de logiciel** qui peut accomplir une tâche prédéfinie spécifique de façon autonome (au nom d'un utilisateur ou d'une application).

Définition d'un agent

- Un agent effectue continuellement trois fonctions :
 - Perception de l'environnement ;
 - Interprétation de ces perceptions ;
 - Détermination des actions possibles.

L'agent dans son environnement

- Un agent dans son environnement, il est activé en entrée par les capteurs de l'environnement et produit en sortie des actions.



Types d'agents

❑ Les agents intelligents

- ❑ Un agent intelligent est une entité logicielle qui réalise des opérations pour le compte d'un usager (ou d'un autre programme) avec un certain degré de **liberté** et **d'autonomie** et qui, pour ce faire, exploite des connaissances ou des représentations des désirs et des objectifs de l'usager.

❑ Les agent mobiles

- ❑ Les agents mobiles sont des entités logicielles qui peuvent se déplacer dans le réseau de leur propre initiative ; ils se déplacent d'une machine à une autre et communiquent avec d'autres agents ou accèdent aux ressources du serveur.

Les agents mobiles

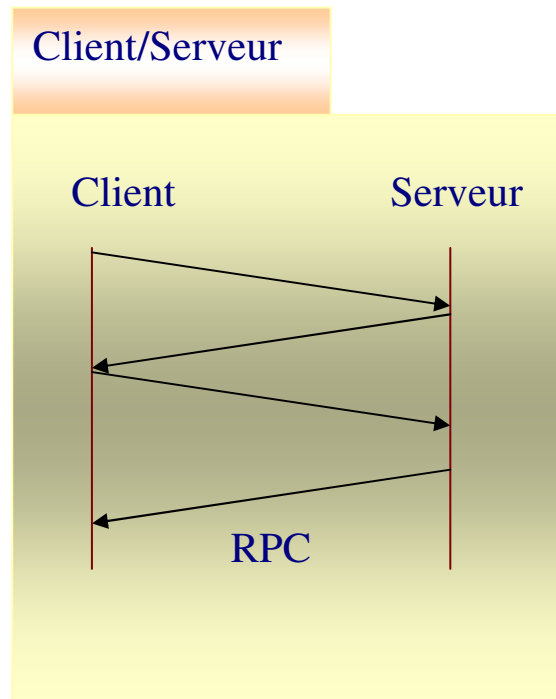
- ❑ La technologie d'agents mobiles permet de fournir une exécution asynchrone de tâche. Ainsi, la dépendance entre les clients et les applications de serveur peut être réduite et un traitement automatique de tâche est présenté.
- ❑ Réaliser une meilleure exploitation de ressource. En transférant des applications du client au serveur et en exécutant des appels locaux de procédure au lieu des appels extérieurs, le trafic réseau est réduit.
- ❑ Les services ne sont plus liés à un certain environnement. Ils peuvent être dynamiquement installés et employés dans l'endroit exact où ils sont exigés.

Les agents mobiles

- ❑ Les principaux attributs d'un agent mobile sont :
 - ❑ La mobilité du **code** ;
 - ❑ La mobilité des **données** ;
 - ❑ La mobilité de **l'état** : prêt/bloqué/exécution réelle.
- ❑ Ceci permet à des entités logicielles de se déplacer de façon autonome à travers le réseau pour accomplir des tâches spécifiques.
 - ❑ Réduire le coût de communication.

RPC vs Agents mobiles

- ❑ L'informatique répartie traditionnelle est généralement basée sur le modèle **client/serveur** qui est fréquemment lié au mécanisme de **RPC (Remote Procedure Call)**.



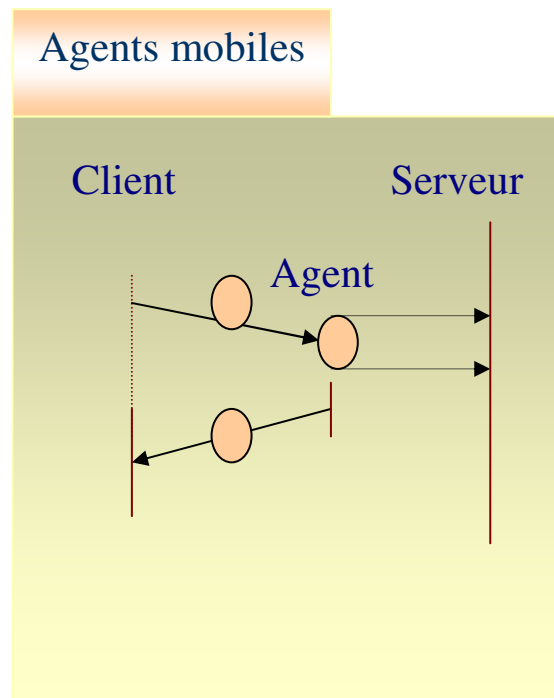
RPC vs Agents mobiles

- ❑ Quand un RPC est appelé, des paramètres sont envoyés au serveur pour exécuter la procédure et le résultat est renvoyé au client.

- ❑ Chaque RPC exige l'établissement d'un canal de communication sur le réseau.

RPC vs Agents mobiles

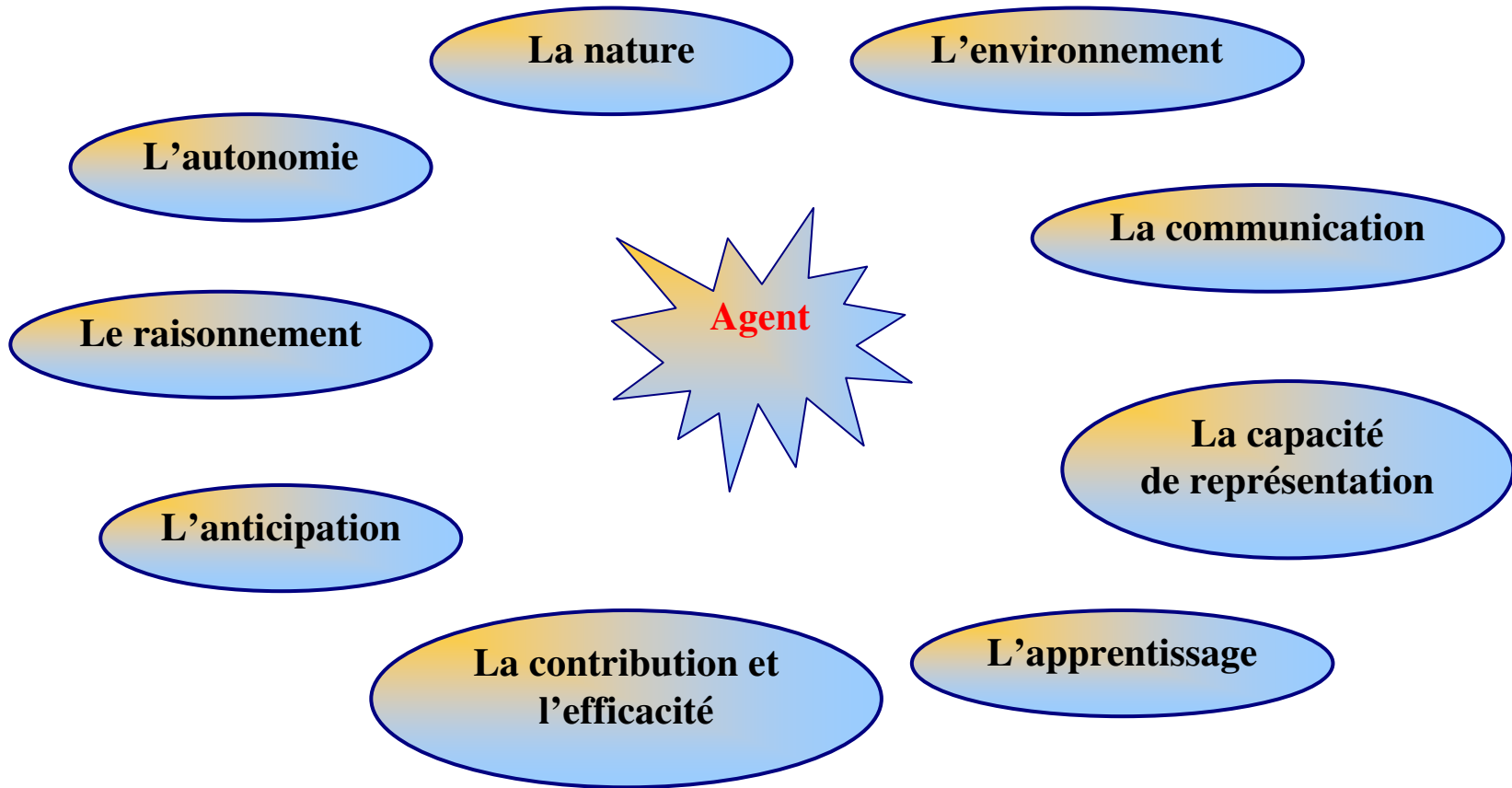
- ❑ Avec les agents mobiles ; **la mobilité de code** permet à une partie du programme client de se déplacer vers le serveur, d'agir localement et de revenir au client, une fois les interactions exécutées.
- ❑ Les améliorations d'exécution dépendent de la bande passante dans le réseau et du nombre de messages qui peuvent être optimisés au passage



Applications des agents mobiles

- ❑ Le commerce électronique ;
- ❑ La recherche dans les bases de données ;
- ❑ La gestion de réseau ;
- ❑ Le calcul mobile.

Les caractéristiques d'un agent



Les caractéristiques d'un agent

□ Un agent est caractérisé par :

- **La nature** : un agent est une entité physique ou virtuelle.
- **L'autonomie** : un agent est indépendant de l'utilisateur et des autres agents.
- **L'environnement** : c'est l'espace dans lequel va agir l'agent, il peut se réduire au réseau constitué par l'ensemble des agents.

Les caractéristiques d'un agent

- ❑ **La capacité de représentation** : l'agent peut avoir une vision très locale de son environnement mais il peut aussi avoir une représentation plus large de cet environnement et notamment des agents qui l'entourent.
- ❑ **La communication** : l'agent aura plus ou moins de capacité à communiquer avec les autres agents.

Les caractéristiques d'un agent

- ❑ **Le raisonnement** : l'agent peut être lié à un système expert ou à d'autres mécanismes de raisonnements plus ou moins complexes.
- ❑ **L'anticipation** : l'agent peut plus ou moins avoir les capacités d'anticiper les événements futurs.

Les caractéristiques d'un agent

- ❑ **L'apprentissage** : un agent aura plus ou moins tendance à retirer, stocker et réutiliser des informations extraites ou reçus de son environnement.
- ❑ **La contribution** : l'agent participe plus ou moins à la résolution de problèmes ou à l'activité globale du système.
- ❑ **L'efficacité** : l'agent doit avoir la rapidité d'exécution et d'intervention.

Architecture concrète pour un agent

**Architecture
d'agents logiques**

Architecture BDI

**Architecture
réactive**

Quatre classes d'agents

- ❑ **Les agents logiques** : les décisions prises par les agents sont basées sur des déductions logiques.
- ❑ **Les agents réactifs** : les décisions prises par les agents sont basées sur une correspondance entre les situations et les actions.
- ❑ **Les agents BDI** : les états internes des agents sont exprimés sous la forme de croyance (Belief), de désirs (Desire) et d'intentions (Intention), la prise de décision est basée sur l'état interne de l'agent.

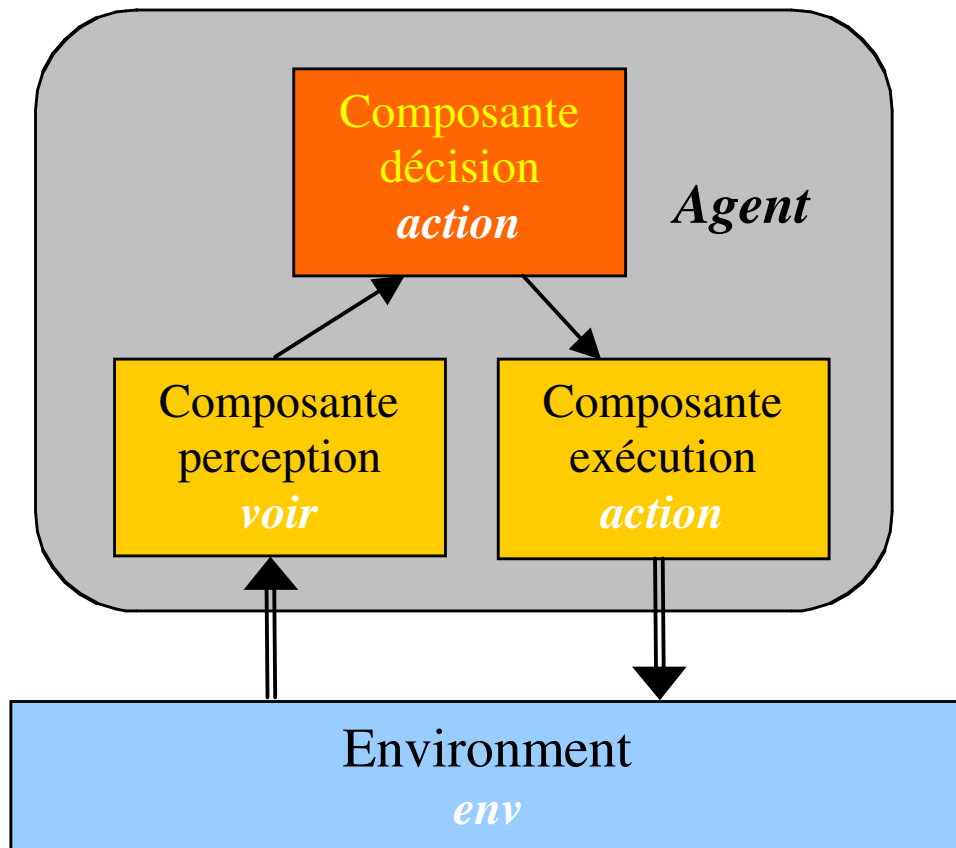
Architecture d'agents logiques

- ❑ Les connaissances des agents sont décrites sous la forme d'expression logique.
- ❑ L'agent utilise la déduction logique pour résoudre les problèmes et pour caractériser son comportement.
- ❑ Dans la pratique, l'approche logique est irréalisable dans des environnements complexes, et fortement dynamiques.

L'architecture réactive

- ❑ Les approches réactives sont issues des problèmes et limites rencontrées par l'approche logique.
- ❑ Les règles de décision sont de la forme **situation** → **action**, elles font correspondre une action à un ensemble de perceptions.

Modélisation



Agent réactif

Architecture BDI

- ❑ Une architecture BDI est composée de trois types d'éléments :
 - ❑ **Les croyances** : les informations que l'agent possède sur son environnement.
 - ❑ Ces croyances peuvent être incorrectes, incomplètes ou incertaines.
 - ❑ **Les désirs** : les états de l'environnement que l'agent aimerait atteindre. Un agent peut avoir des désirs contradictoires dans ce cas, il choisit un sous ensemble consistant de ses désirs.
 - ❑ Cet sous-ensemble correspond aux buts de l'agent.
 - ❑ **Les intentions** : ce sont les actions que l'agent a décidé d'accomplir pour atteindre ses buts.

Architecture BDI

- ❑ Au cours de l'exécution de ses actions, l'agent peut juger que ses intentions ne sont plus pertinentes (modification des croyances, impossibilité d'atteindre les buts ou d'exécuter les actions résultant des intentions...).
- ❑ L'agent doit alors mettre en place des mécanismes lui permettant de réviser ses croyances, ses désirs et ses intentions.

Architecture BDI

- ❑ Formellement un agent BDI est caractérisé par :
 - ❑ **Bel** : l'ensemble des croyances possibles.
 - ❑ **Des** : l'ensemble des désirs possibles.
 - ❑ **Inten** : l'ensemble des intentions possibles.
- ❑ L'état de l'agent est décrit à tout moment par le triplet (B,D,I) où $B \subseteq \text{Bel}$, $D \subseteq \text{Des}$, $I \subseteq \text{Inten}$.

Architecture BDI

- ❑ Chaque agent BDI est décomposé en au moins sept niveaux fonctionnels, ces niveaux sont les suivants :
 - ❑ Un ensemble de croyances courantes (B) sur son environnement.
 - ❑ Une fonction de révision de ses croyances (brf) qui calcule ses nouvelles croyances à partir des croyances courantes et de ses nouvelles perceptions de son environnement.
 - ❑ Un ensemble de désirs (D).
 - ❑ Une fonction de génération de ses options pertinentes (options) qui représentent ses désirs possibles conformément à ses intentions.

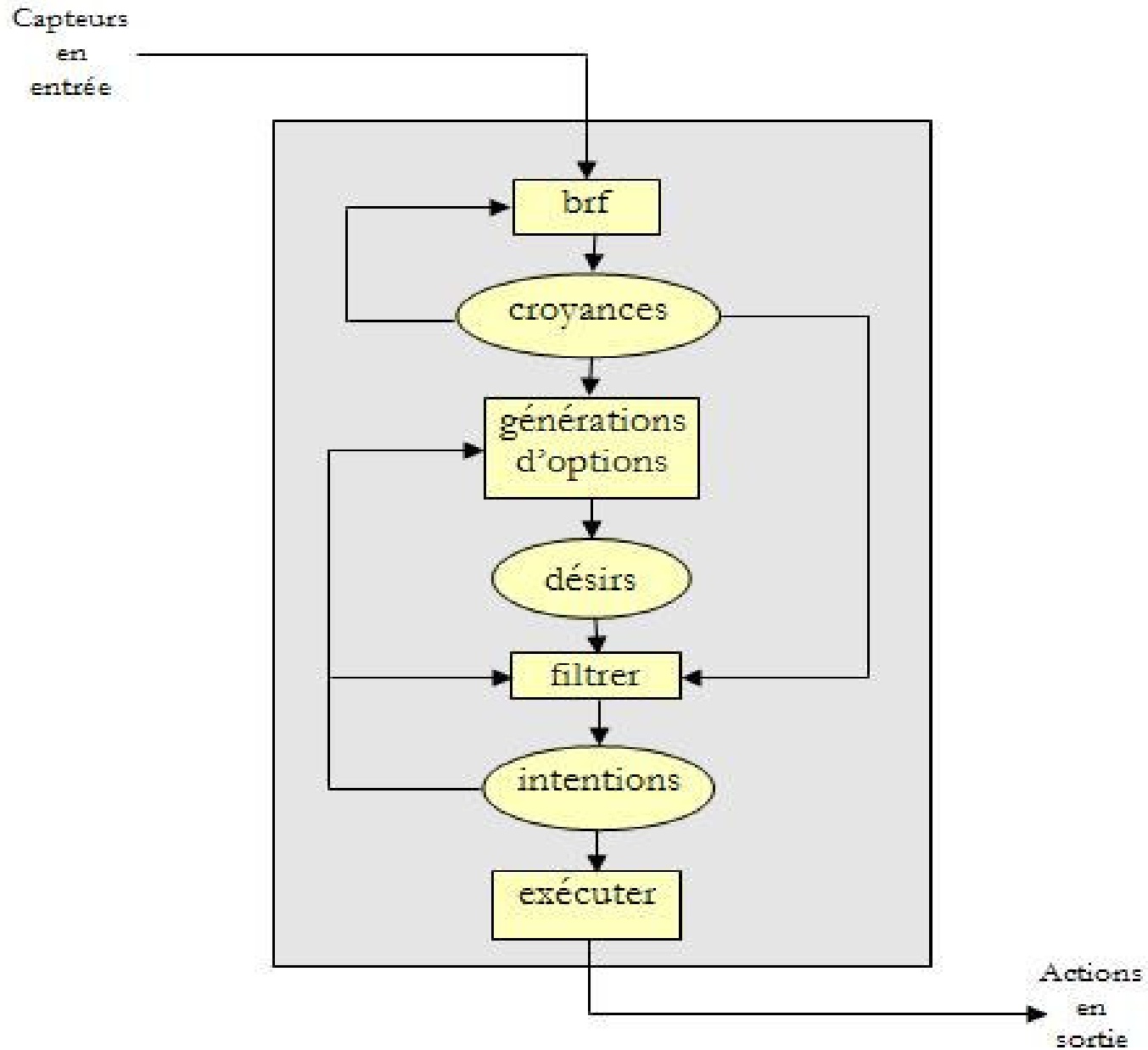
Architecture BDI

- ❑ Un ensemble d'**intentions courantes (I)**, représentant ses centres d'intérêts actuels.

- ❑ Une **fonction de filtre (filtre)** qui active ses nouvelles intentions en fonction de ses croyances, options, et intentions courantes.
 - ❑ Cette fonction élimine les intentions devenues irréalistes ou incohérentes.

- ❑ Une fonction de sélection (**exécuter**) de l'action à exécuter, cette fonction renvoie une intention exécutable qui correspond à une action.

Architecture d'un agent BDI



Modèle type d'un agent

- ❑ Une architecture implémentée avec trois niveaux de description :
 - ❑ Le 1er niveau (réactif)
 - ❑ Le 2ème niveau (adaptatif)
 - ❑ Le 3ème niveau (cognitif)

Modèle type d'un agent

- ❑ Le 1er niveau (réactif) : **situation** → **action**, n'a pas de capacité d'apprentissage.
- ❑ Le 2ème niveau (adaptatif) : capable de modifier son comportement et ses objectifs en fonction des interactions.
- ❑ Le 3ème niveau (cognitif) : raisonnement.

Modèle type d'un agent

- Les trois niveaux de description :

Le 1^{er} niveau (réactif)

Le 2^{ème} niveau (adaptatif)

Le 3^{ème} niveau (cognitif)

Bibliographie

- ❑ M.N. Huhns. “Multi-agent Systems”. Tutorial at the European Agent Systems Summer School, (EASSS'99), 1999.
- ❑ T. Magedanz, K. Rothermel and S. Krause. “Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?”. INFOCOM'96, San Francisco, CA, USA, March 1996.
- ❑ M. Breugst and T. Magedanz. “Mobile Agents – Enabling Technology for Active Intelligent Network Implementation”. IEEE Network, 53--60, May/June 1998.
- ❑ D. Kotz and R. Gray. “Mobile Agents and the Future of the Internet”, ACM Operating Systems Review, Vol. 33(3), pp. 7-13, 1999.
- ❑ B. Benmammam et F. Krief. "La Technologie Agent et les Réseaux Sans Fil". Dans les actes du 17^{ème} Congrès Des Nouvelles Architectures pour les Communications. DNAC'2003. Paris, France. Octobre 2003.
- ❑ B. Benmammam and F. Krief. "Agents for Wireless Environments". Proceedings of the International Conference on Telecommunication Systems, Modeling and Analysis. ICTSM'2004. IFIPWG 7.3. Monterey, USA. July 2004. ISBN: 0971625352.

Bibliographie

- ❑ Zeina Jrad, Badr Benmammam, Joseph Corr ea, Francine Krief, Nader Mbarek. "A User Assistant for QoS Negotiation in a Dynamic Environment Using Agent Technology". Proceedings of the Second IEEE and IFIP International Conference on Wireless and Optical Communications Networks WOCN 2005. March 6 - 8, 2005, Hyatt Regency Hotel, Dubai, United Arab Emirates UAE, Print ISBN: 0-7803-9019-9.
- ❑ B. Benmammam, Z. Jrad and F. Krief. "QoS management in mobile IP networks using a terminal assistant". Wiley/ACM International Journal of Network Management, Wiley InterScience Edition, ISSN (printed): 1055-7148, ISSN (electronic): 1099-1190, Volume 19, Issue 1, Date: January/February 2009, Pages: 1-24.
- ❑ B. Benmammam and F. Krief. "MQoS NSLP: a mobility profile management based approach for advance resource reservation in a mobile environment". Proceedings of the 7th IFIP IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2005). Marrakech, Morocco. September 19-21, 2005. ISBN: 0-955 1814-0-2.
- ❑ Z. Jrad, F. Krief and B. Benmammam. "An Intelligent User Interface for the Dynamic Negotiation of QoS". Proceedings of the 10th IEEE International Conference on Telecommunications.ICT'2003. Papeete, Tahiti. February 2003, Print ISBN: 0-7803-7661-7.