

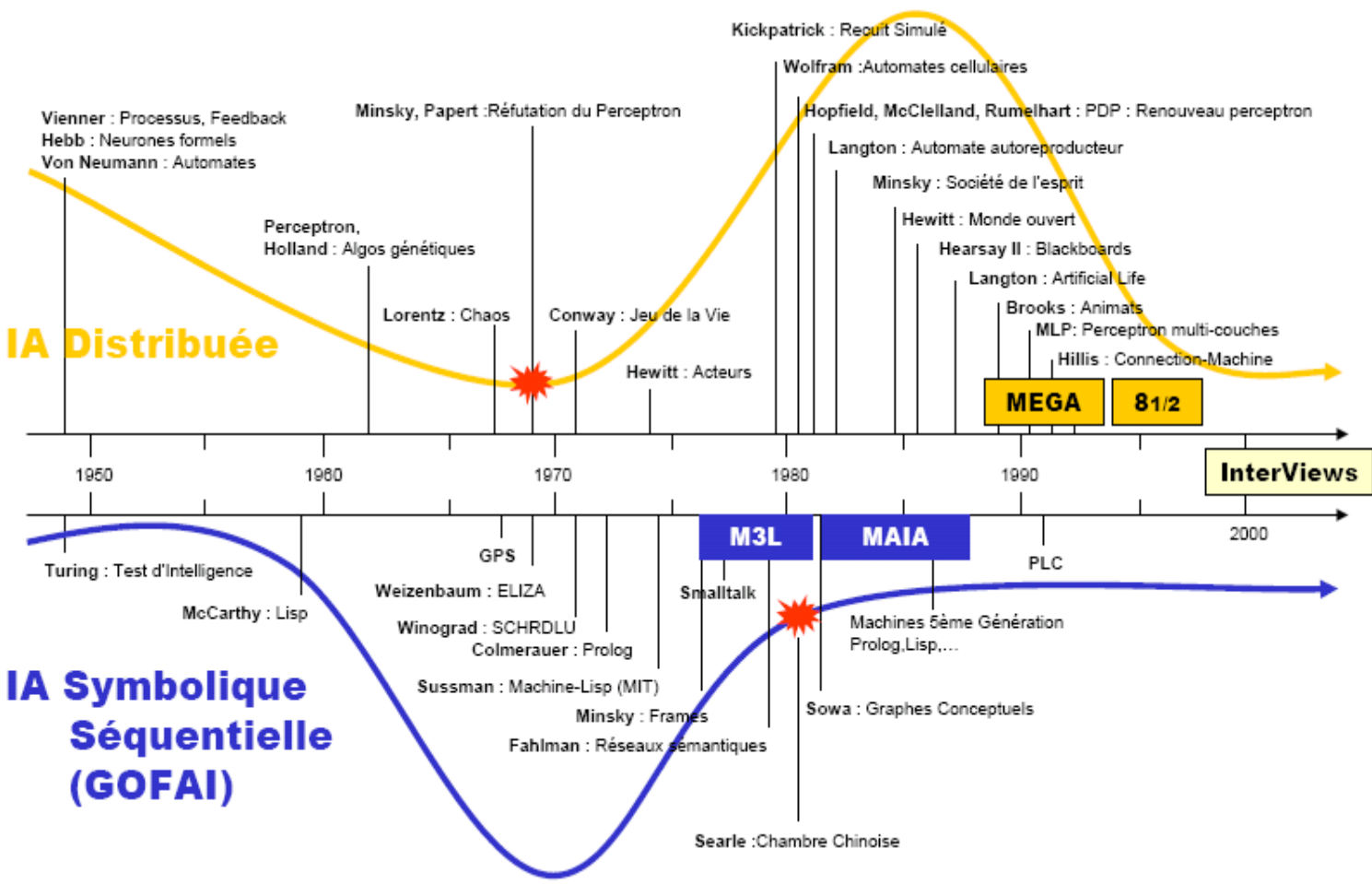
# Intelligence Artificielle Agents Intelligents

## Plan

- Motivations
- Qu'est-ce qu'un agent intelligent?
- Comment les agents intelligents doivent agir?
- Structure d'un agent intelligent
- Types d'environnements
- Conclusion
- Références

# Motivations

# Motivations



# Motivations

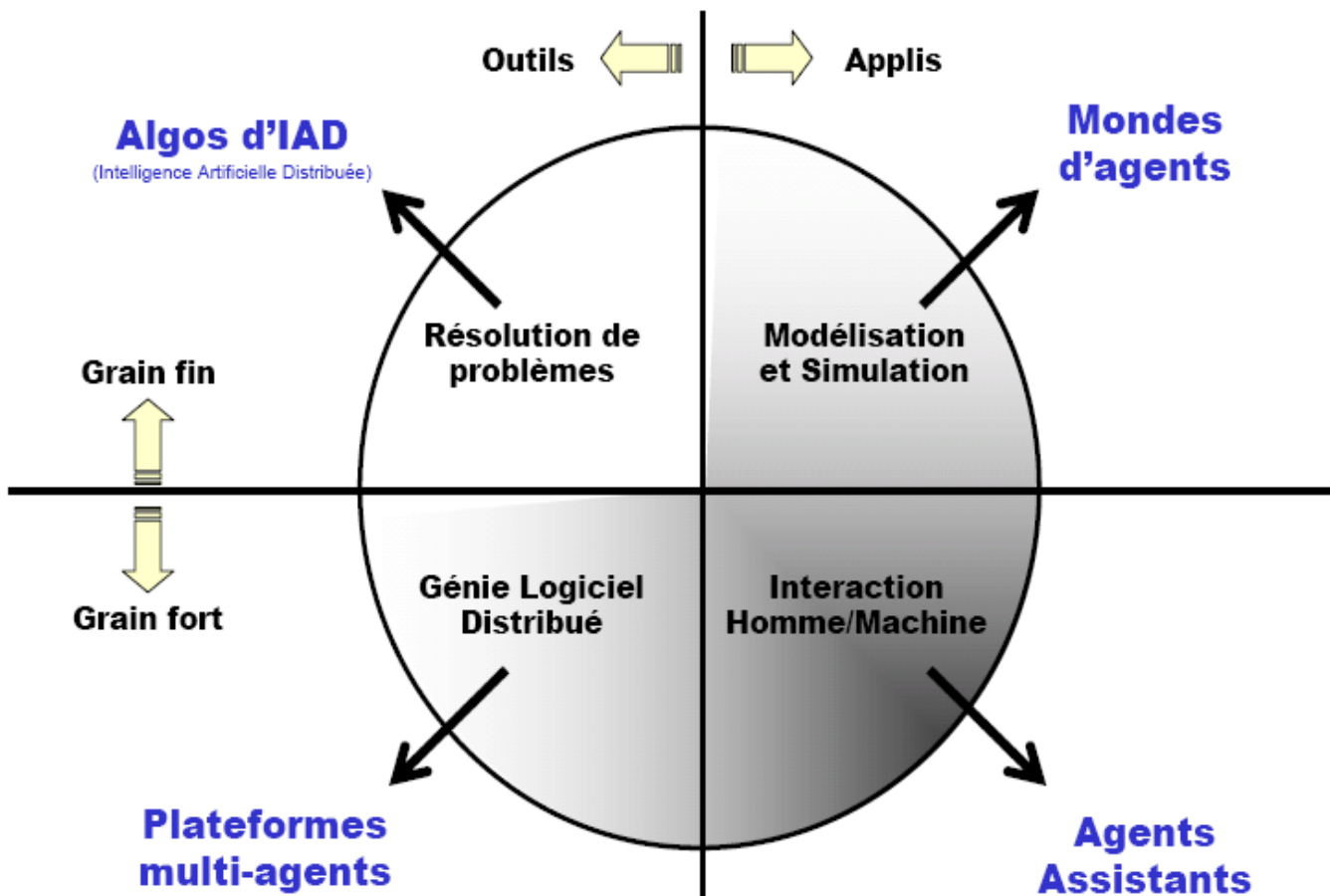
**IDC (<http://www.idc.com>)**  
**(IDC de : International Data Group)**

**IDC estime le marché global pour les agents (logiciel):**

- **7.2 millions de dollars 1997**
- **51.5 millions dollars en 1999**
- **837 millions dollars en 2004**

# Motivations

- **Systèmes repartis complexes**: comprendre, construire, gérer
- **Systèmes ouverts et hétérogènes**: construction des composantes, interaction des composantes
- **Distribution des ressources**
- **Distribution de l'expertise**
- **Interopérabilité** avec les systèmes existants /intégration avec le logiciel classique (legacy systems)



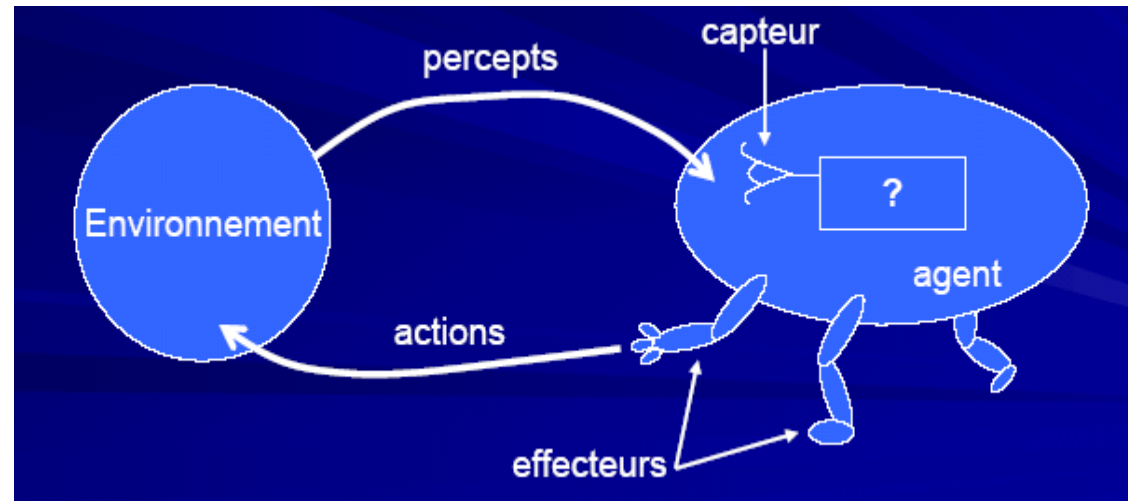
**Qu'est-ce qu'un agent intelligent?**



# Qu'est-ce qu'un agent intelligent?

## Une définition

**Agent** : tout ce qui **perçoit** son **Environnement** à l'aide de ses **capteurs** et qui **agit** sur son environnement l'aide de ses **effecteurs**..



### Pour un humain:

- **Capteurs** : oreilles, yeux, nez, peau, ...
- **Effecteurs** : mains, bras, jambes, bouche, ...

### Pour un robot :

- **Capteurs** : caméra vidéo, laser, capteur infra-rouge, tactile, ...
- **Effecteurs** : bras robotisé, roues, pinces, ...

### Pour un agent logiciel (softbot) :

- **Capteurs** : données, chaînes de caractères, ...
- **Effecteurs** : données, chaînes de caractères, ...

[Robot-exemple](#)

Agent – il n'y a pas une définition acceptée en unanimité

- Un agent est un système informatique, **situé** dans un environnement, et qui agit d'une façon **autonome** pour atteindre les **objectifs** (buts) pour lesquels il a été conçu (Wooldrige et Jennings, 1995).
- Les agents intelligents sont des entités logiciels qui ***réalisent des opérations à la place d'un utilisateur*** ou d'un autre programme, avec une sorte d'indépendance ou **d'autonomie**, et pour faire cela ils utilisent une sorte de connaissance ou de représentation des buts ou des désires de l'utilisateur." (L'agent IBM);

Agent – il n'y a pas une définition acceptée en unanimité

- Un agent est une entité qui fonctionne continuellement et de manière **autonome** dans un environnement où **d'autres processus** se déroulent et **d'autres agents** existent.« (Shoham, 1993);
- Un agent est **une entité autonome**, réelle ou abstraite, qui est capable **d'agir** sur elle-même et sur son environnement, qui, dans un **univers multi-agents**, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents (Ferber, 1995).

Agent – il n'y a pas une définition acceptée en unanimité

MAIS....

### Traits principaux:

un système informatique (logiciel et/ou matériel)

- situé dans un **environnement**
- capable d'**action autonome** dans cet environnement
- pour satisfaire ses **objectifs** de conception
- possédant des **caractéristiques** agent

« Intelligence » = **flexibilité du comportement**

**Réactivité** par rapport à l'environnement

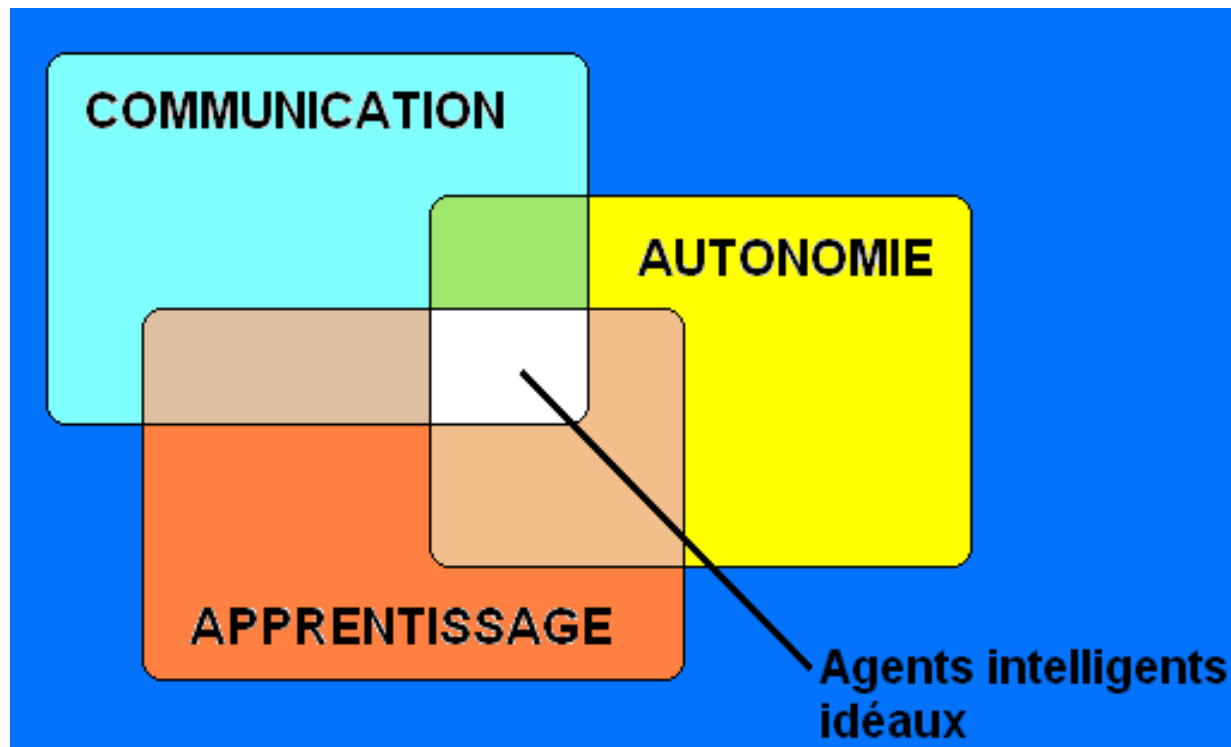
**Pro-activité**: comportement visant la réalisation de **buts**

**Capacité sociale**: coopération avec d'autres agents  
(logiciels ou humains) pour réaliser les buts

# Caractéristiques d'un agent

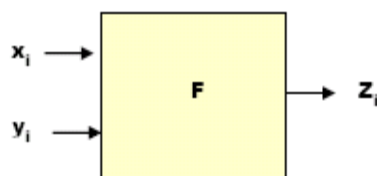
- **situé** – l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement;
- **autonome** – l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne;
- **proactif** – l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment;
- **Mobile**: capable de se déplacer vers un autre environnement
- **Rational**: capable d'agir en fonction de ses objectifs interne et ses connaissances
- **Intelligent**: Conscient de son raisonnement: croyances, objectifs, plans, hypothèses sont explicites
- **social** – l'agent doit être capable d'interagir avec d'autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs: Coopération, concurrence, .....
- .....

# Caractéristiques d'un agent (L'essentiel)



# L'environnement: Le Contexte

## Fonction : pas de contexte



Une fonction est une entité passive (il faut l'appeler pour qu'elle agisse) qui n'a pas de « dehors » :

- Elle ne perçoit pas d'extérieur (normalement elle n'affecte pas de variables globales),
- Elle n'a pas d'histoire (chaque exécution est indépendante des précédentes)
- Elle n'agit pas sur les choses. Il n'y pas d'effet de bord ni sur les arguments  $x, y$  (calculer  $z$  ne modifie pas  $x$  et  $y$ ) ni sur les variables globales

### BAD !

```
F[x_] := (CPT++; x+CPT)
```

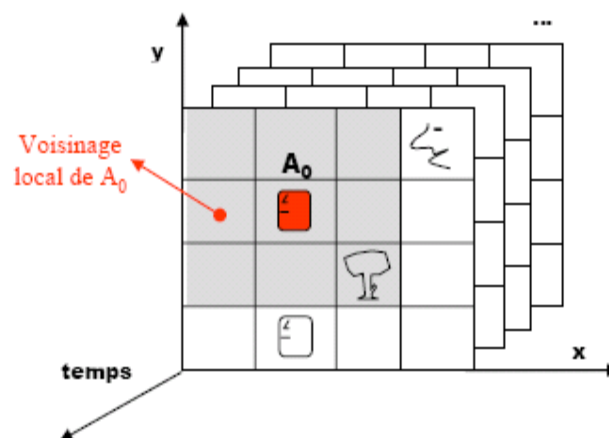
```
CPT=1; Print[f[1]] ; Print[f[2]]
```

```
⇒ 3
```

```
⇒ 5
```

La variable globale CPT peut être vue comme un environnement minimaliste pour  $f$  qui n'est plus une pure fonction.

## Agent : agir en contexte



Un agent  $A_0$  est situé en  $(x,y)$  dans un **espace** et il possède un **voisinage** :

- Il peut le **percevoir localement** et non globalement
- Il peut se le **représenter**
- Il peut **agir** sur cet environnement :
  - Relativement : par exemple en se déplaçant
  - Absolument : par modification de l'état physique ou mental des choses du voisinage (choses = des objets, d'autres agents, des humains)
- Il existe une dimension **temporelle** ( $t$ ) : Chaque changement d'état de l'agent et/ou des choses modifie l'environnement et fait progresser le temps.

# Agents vs Objets

**Agent:** entité autonome interagissant avec son environnement

**Objet:** entité passive possédant un état et sur lequel on peut effectuer des opérations.

Un agent peut être constitué de plusieurs objets.

C'est un paradigme de programmation mettant en évidence l'autonomie et les interactions. (Programmation orientée-agent)

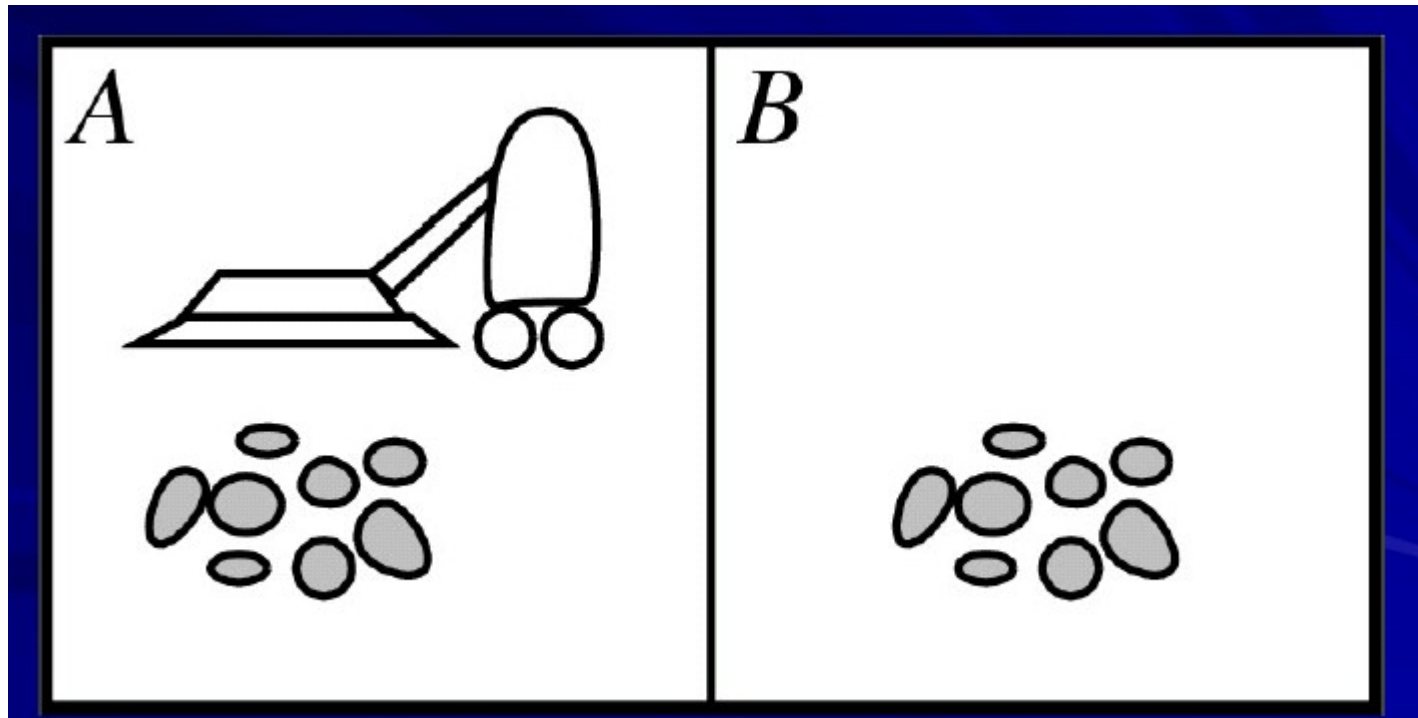


# Agents vs Objets

| <b>Objets</b> | <b>Agents</b>  |
|---------------|--|
| Classe        | Rôles, Types   |
| Message       | Assertion, Requête,<br>Commande, Changement<br>d'état, Événement |
| Attribut      | Etat, Croyances, Objectifs,<br>Meta connaissance                 |
| Méthode       | Plans, Capacités, Règles,<br>Services, Responsabilités           |

# Exemple très simple d'agent

Un agent aspirateur



| Séquence de perception           | Action |
|----------------------------------|--------|
| [A,Propre]                       | Droite |
| [A,Sale]                         | Aspire |
| [B,Propre]                       | Gauche |
| [B,Sale]                         | Aspire |
| [A,Propre],[A,Propre]            | Droite |
| [A,Propre],[A,Sale]              | Aspire |
| :                                | :      |
| [A,Propre],[A,Propre],[A,Propre] | Droite |
| :                                | :      |

# Agent rationnel

**Rationnel = basé sur le raisonnement**

**Agent rationnel:** l'agent doit exécuter l'action qui maximise sa mesure de performance en fonction de sa perception du monde et de ses connaissances.

*« For each possible percept sequence, an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has. »*

Russel, Norvig, page 33.

Mesure de performance:

- Réussir une tâche
- Quantité de ressources consommées
- Temps mis pour réaliser la tâche

## Agent rationnel

Rationnel (ce n'est pas).

- Omniscient
- Clairvoyant
- Succès

Rationnel (c'est)

- Exploration
- Apprentissage
- Autonomie

La notion d'agent est un outil d'analyse de systèmes et non pas une caractérisation absolue qui divise le monde en de deux : Agents et non-Agents

## Choix de la mesure de performance

( ex: l'agent Aspirateur)

Il faut faire attention à ce qu'on choisit.

–Ex: Quantité de saleté ramassée en 8 heures, bruit généré, consommation d'électricité,....

Une meilleure option:

– Récompenser l'agent pour un plancher propre

Ex: un point pour chaque carré propre à chaque intervalle de temps  
(Peut-être avec une pénalité pour l'électricité consommé).

## Structure d'un agent

**Agent = architecture + programme:**

- *Programme*: une procédure qui met en correspondance les séquences perceptives et les actions
- *Architecture*:
  - \* Organisation matérielle et logicielle
  - \* Contrôle les capteurs et « fabrique » les séquences perceptives
  - \* Exécute les programmes et les actions ( contrôle les effecteurs)

Pour Chaque agent il faut analyser :

sa séquence perceptive (Percepts), ses actions (Actions), ses objectifs (Goals) et son environnement (Environment) = PAGE

## PAGE: Chauffeur de taxi

**Percepts:** Caméra vidéo, capteurs moteur, GPS, accéléromètre,...

**Actions:** accélérer, freiner, braquer, klaxonner, discuter,...

**Goals:** Atteindre la destination, respecter les règles de circulation, sécurité,.

**Environment :** rues, routes, piétons, clients, condition météo,...



# Structure d'un agent

## Agents cognitifs

Pour appliquer le modèle de l'intelligence humaine et la perspective humaine sur le monde, il est courant dans la communauté des chercheurs d'intelligence artificielle de caractériser un agent intelligent en utilisant des *notions mentales* telles que la **connaissance, la croyance, les intentions, les désirs, les choix, les engagements** (Shoham, 1993).

### Exemples des notions mentales:

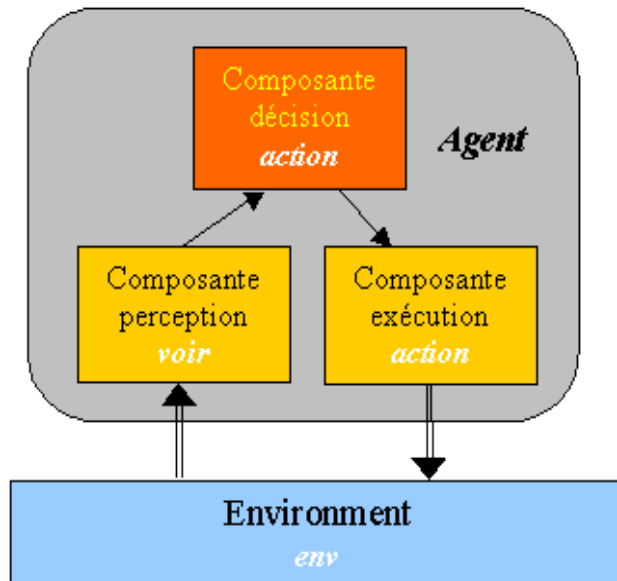
|                         |   |
|-------------------------|---|
| <b>connaissances</b>    | Moha connaît le fait que les humains sont mortels                                 |
| <b>croyances</b>        | Moha a pris son parapluie parce qu'il croit qu'il va pleuvoir                     |
| <b>désirs, buts</b>     | Moha désire avoir son doctorat  |
| <b>intentions</b>       | Moha a l'intention de travailler dur pour avoir sa thèse                          |
| <b>choix, décisions</b> | Moha a décidé de faire une thèse  |
| <b>engagements</b>      | Moha ne va pas s'arrêter de travailler avant d'avoir fini sa thèse                |
| <b>conventions</b>      | si, par hasard, Moha décide d'abandonner sa thèse, il va le dire à son professeur |
| <b>obligations</b>      | Moha doit travailler pour entretenir sa famille                                   |

## Agents réactifs

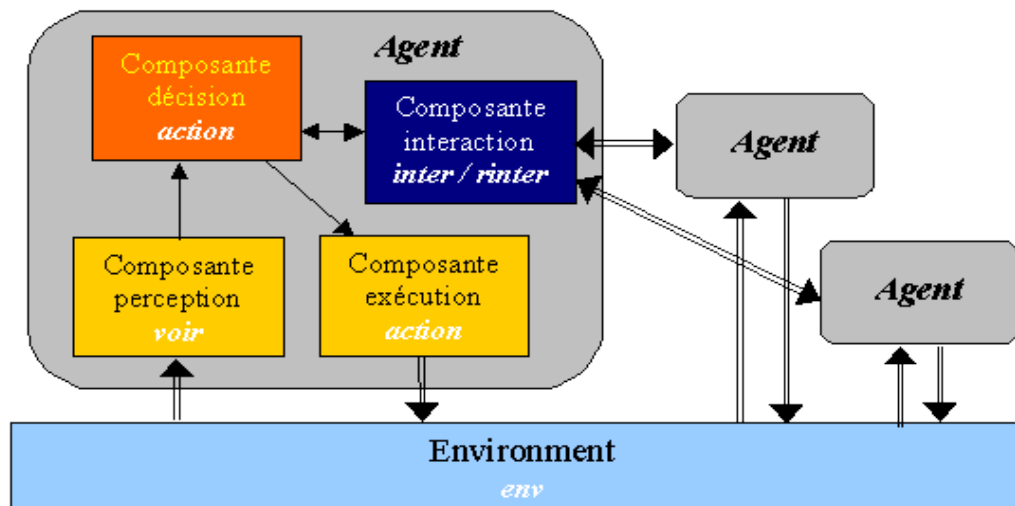
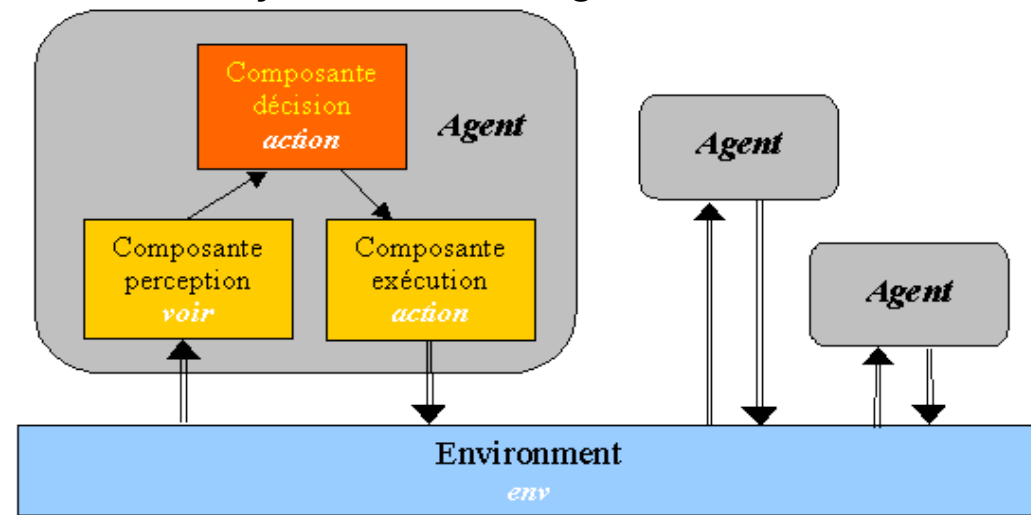
Les agents réactifs sont souvent qualifiés de ne pas être “ intelligents ” par eux-mêmes. Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent pas de croyances, pas de mécanismes d'envoi de messages. Leurs capacités répondent uniquement au mode stimulus/action qui peut être considéré comme une forme de communication. **EX: fonction mathématique**

# Structure d'un agent

Structure générale d'un agent



Structure générale d'un agent dans un système multi-agents



Structure générale d'un agent qui interagit avec d'autres agents

## Structure d'un agent

4 types de base en ordre de généralité grandissante:

- Agent simple réflexe
- Agent réflexe avec état interne

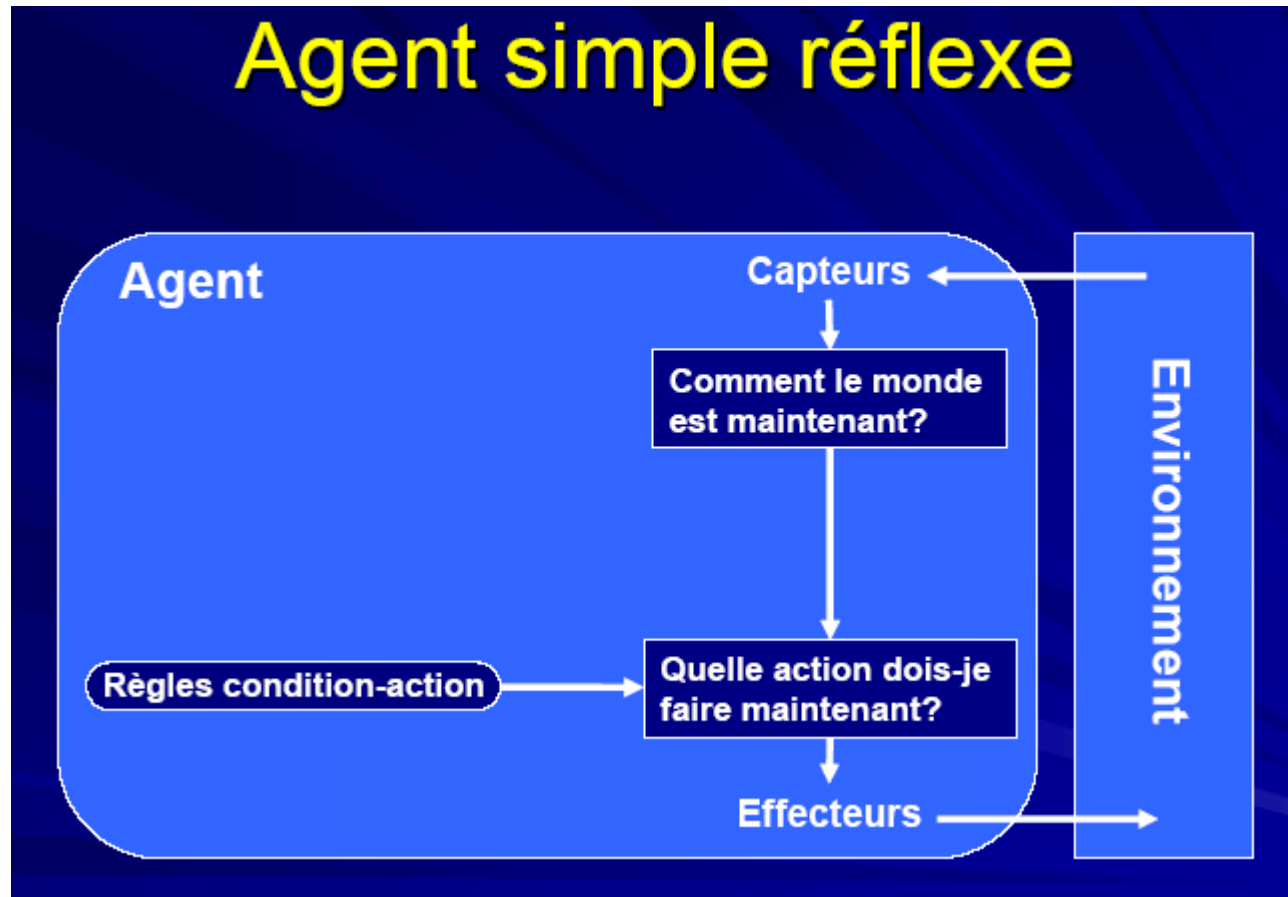
Réactif ↑

- Agent basé sur les buts
- Agent basé sur l'utilité

Cognitif ↓

Tous ces types peuvent être transformés en agents apprenants.

# Agent simple réflexe



# Agent simple réflexe

Ce type d'agent choisit ses actions en se basant uniquement sur le percept courant, en ignorant les percepts précédents.

**Action** **Reflex-simple( Percept percept)**

*{ ---Déclarations*

**Static Regle** **regles[]**; *---liste de conditions-actions : si le feu est vert alors passer*

**Regle** **regle**;

**Action** **action**;

**Etat** **etat**;

*---Executions*

**etat** = **input-interpreter(percept)**;

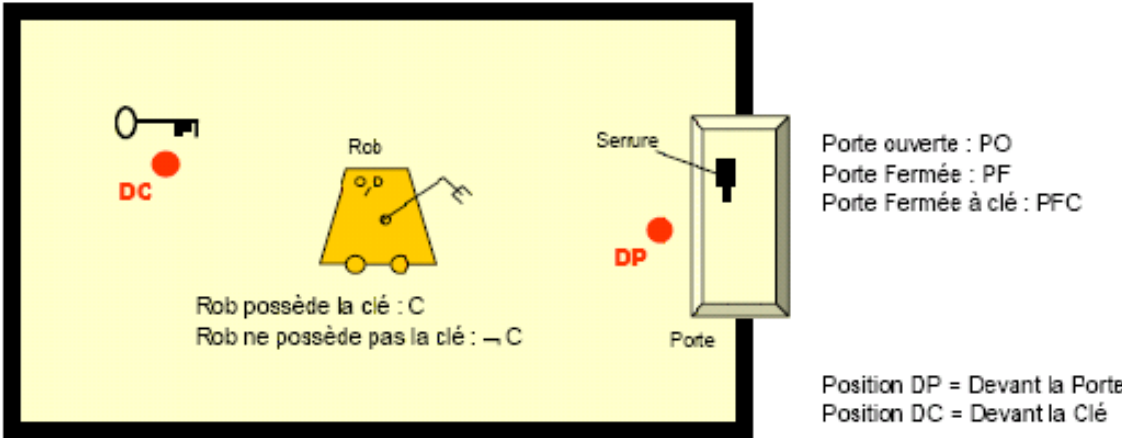
**regle**=**regle-correspondre(etat,regles)**

**action** = **Regle-Action(regle)**; *// recherche action selon la règle en cours*

**return action**;

**}**

Problème : « Rob doit sortir de la pièce ».



**Rob réactif**

la **fonction de transition** T est un ensemble non ordonné de règles de type  
Si condition alors action  
qui sont exécutées dans une boucle infinie et non déterministe :

|                 |                    |
|-----------------|--------------------|
| Si DP ∧ PO      | Alors SORTIR       |
| Si DP ∧ PF      | Alors OUVRIR       |
| Si DP ∧ PFC ∧ C | Alors DEVEROUILLER |
| Si DP ∧ PFC     | Alors RANDOM-WALK  |
| Si DC ∧ C       | Alors RANDOM-WALK  |
| Si DC ∧ ¬ C     | Alors PRENDRE-CLE  |

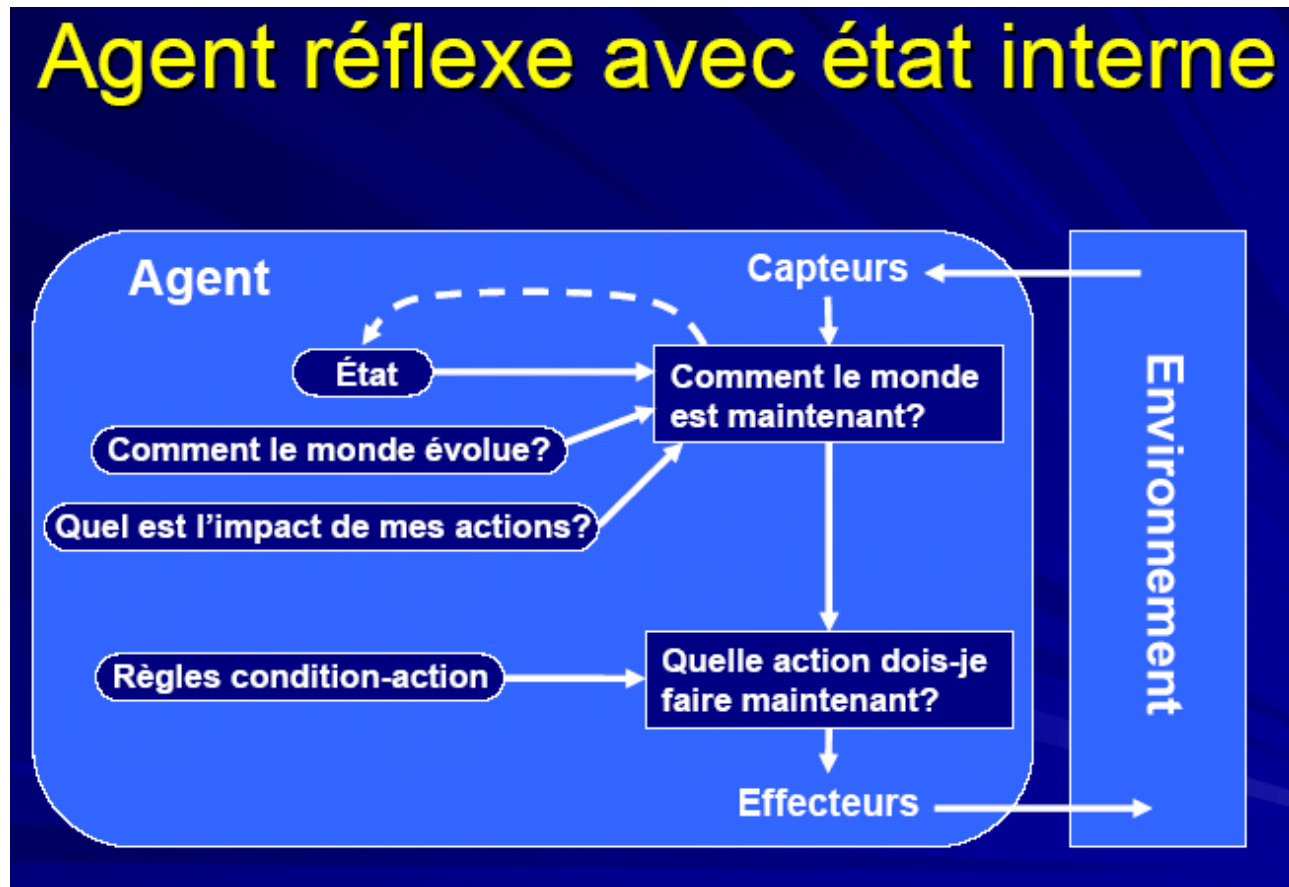
Question : prouver que le système {règles + CI} où CI = conditions initiales quelconques, **converge** c'est-à-dire que Rob finit bien par sortir au bout d'un certain temps ».

**Rob cognitif**

L'agent possède un **plan** P pour sortir qu'il exécute de manière séquentielle et déterministe quelle que soit sa position de départ et quel que soit l'état initial du monde :

ALLER-A DP — appel du sous-plan ALLER-A(x,y)  
Si PO Alors SORTIR  
Sinon Si PF Alors OUVRIR; SORTIR  
Sinon Si PFC  
    Alors Si C Alors DEVEROUILLER; OUVRIR; SORTIR  
    Sinon ALLER-A DC  
        PRENDRE-CLE  
        ALLER-A DP  
        DEVEROUILLER; OUVRIR; SORTIR

Etat interne est utilisée pour distinguer entre les états de l'environnement.  
Nécessité de représenter les changements de l'environnement ( effets des actions)



Etat interne: Pourquoi c'est utile ?

# Agent simple réflexe avec Etat

Etat interne est utilisée pour distinguer entre les états de l'environnement.

Nécessité de représenter les changements de l'environnement ( effets des actions)

**Action** **Reflex-simple-Etat( Percept percept)**

*{ ---Déclarations*

**Static Regle** **regles[];** *---liste de conditions-actions : si le feu est vert alors passer*

**Etat** **etat;**

**Regle** **regle;**

**Action** **action;**

*---Executions*

**etat = Mise-a-jour(etat,percept);**

**regle=regle-correspondre(etat,regles)**

**action = Regle-Action(regle);** *// recherche action selon la règle en cours*

**etat = Mise-a-jour(etat,action);**

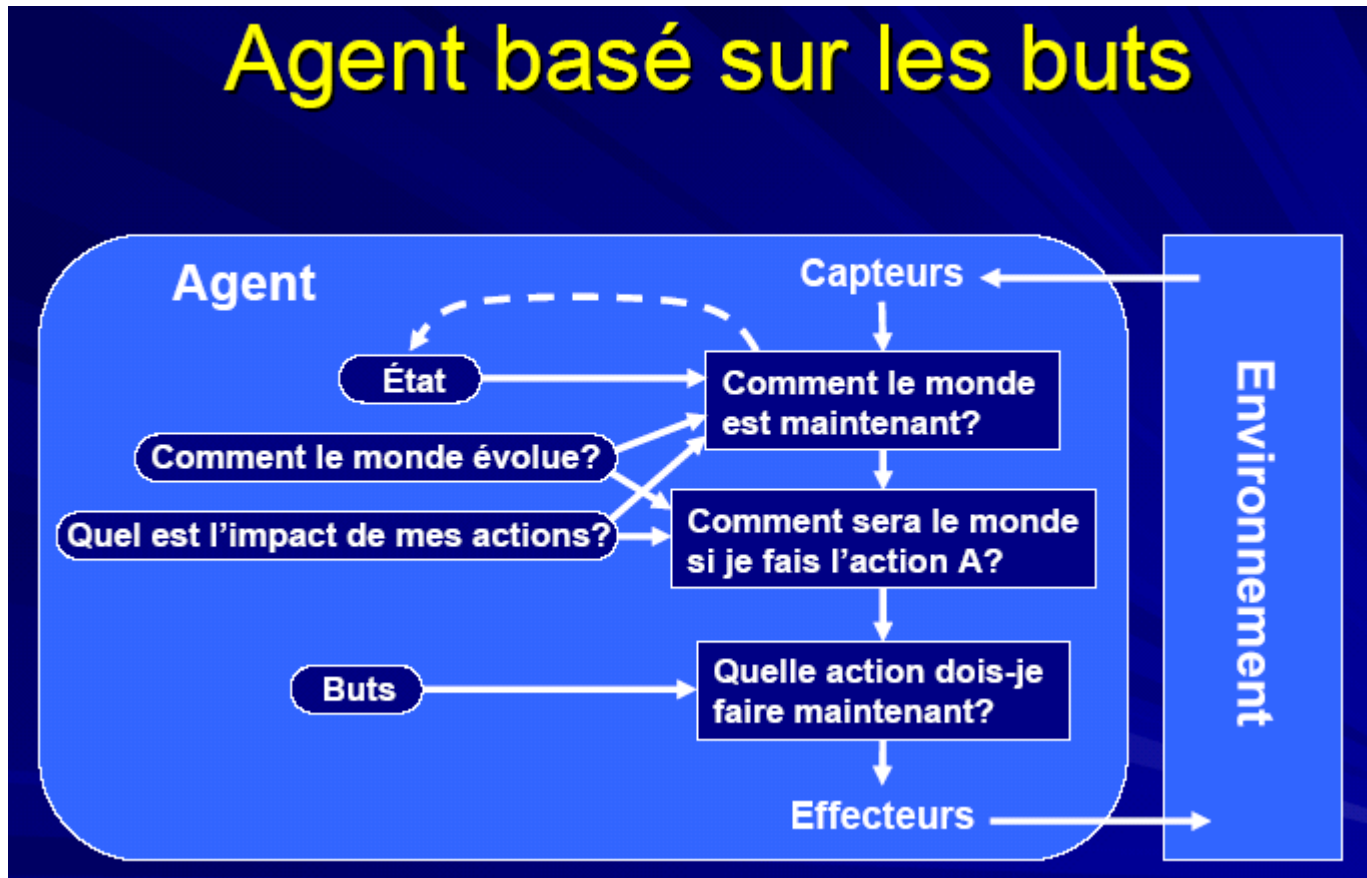
**return action;**

**}**



Résout un problème posé

## Agent basé sur les buts



# Agent Basé sur les Buts

·  
Résout un problème posé

**Action** Agent-a buts( Percept percept)

{ *//---Déclarations*

**Probleme** probleme;

**But** but;

**Etat** etat;

**Action** action;

*//---Executions*

**etat** = Mise-a-jour(etat,percept);

**action** = Rechercher-But(etat); *// recherche action selon l'état en cours*

**etat** = Mise-a-jour(etat,action);

**return** action;

}

Dans les architectures à base de Règles, on assigne un système expert (ou système de règles) à un seul agent. Un système expert est composé de trois parties essentielles :

- Une base de faits (BF) qui contient la description des états physiques et mentaux du monde,
- Une base de règles (BR) qui permet de déduire des faits à partir d'autres faits,
- Un moteur d'inférence : ensemble d'heuristiques de planification qui utilisent BF et BR pour effectuer une tâche donnée.

#### Base de faits (BF) :

$\exists$  T en 1  
 $\exists$  C en 2  
 $\exists$  GC en 3  
 $\exists$  B en 3  
 $\text{upon}[B, GC]$

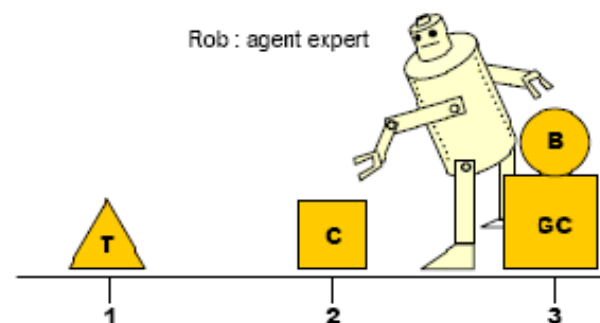
#### Base de règles (BR) :

Les règles sont de la forme : If <condition-list> Then <action-list>

Exemple : un plan en trois règles pour Poser T sur GC.

```

If free(T)  $\wedge$  free(GC) then PUTON(T,GC)
If  $\neg$  free(T) Then MOVETO(X|upon(T,X), Y|free(Y))
If  $\neg$  free(GC) Then MOVETO(X|upon(GC,X), Y|free(Y))
  
```



**Le monde de cubes**

Terry Winograd (1972)

On a les conventions suivantes :

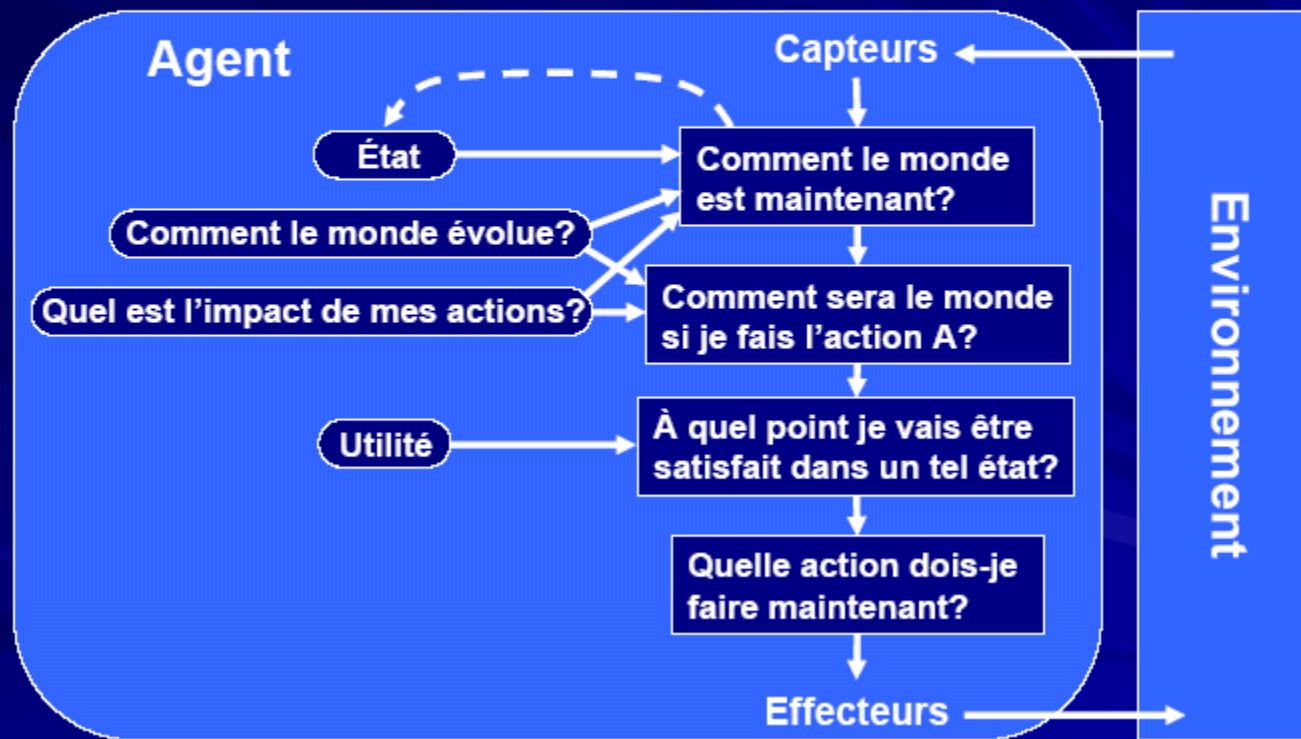
- X, Y sont des variables sur {T, C, GC, B}
- MOVETO et PUTON sont d'autres plans
- free et upon sont des fonctions

## Agent basé sur l'utilité

Les buts ne font la distinction que entre état «heureux» ou «pas heureux»..

- \* Fonction d'utilité:  $P: \text{état} \rightarrow \text{nombre}$
- \* Aide dans deux cas où les buts échouent:
  - Buts en conflits (ex: vitesse et sécurité)
  - Lorsqu'il y a plusieurs buts

# Agent basé sur l'utilité



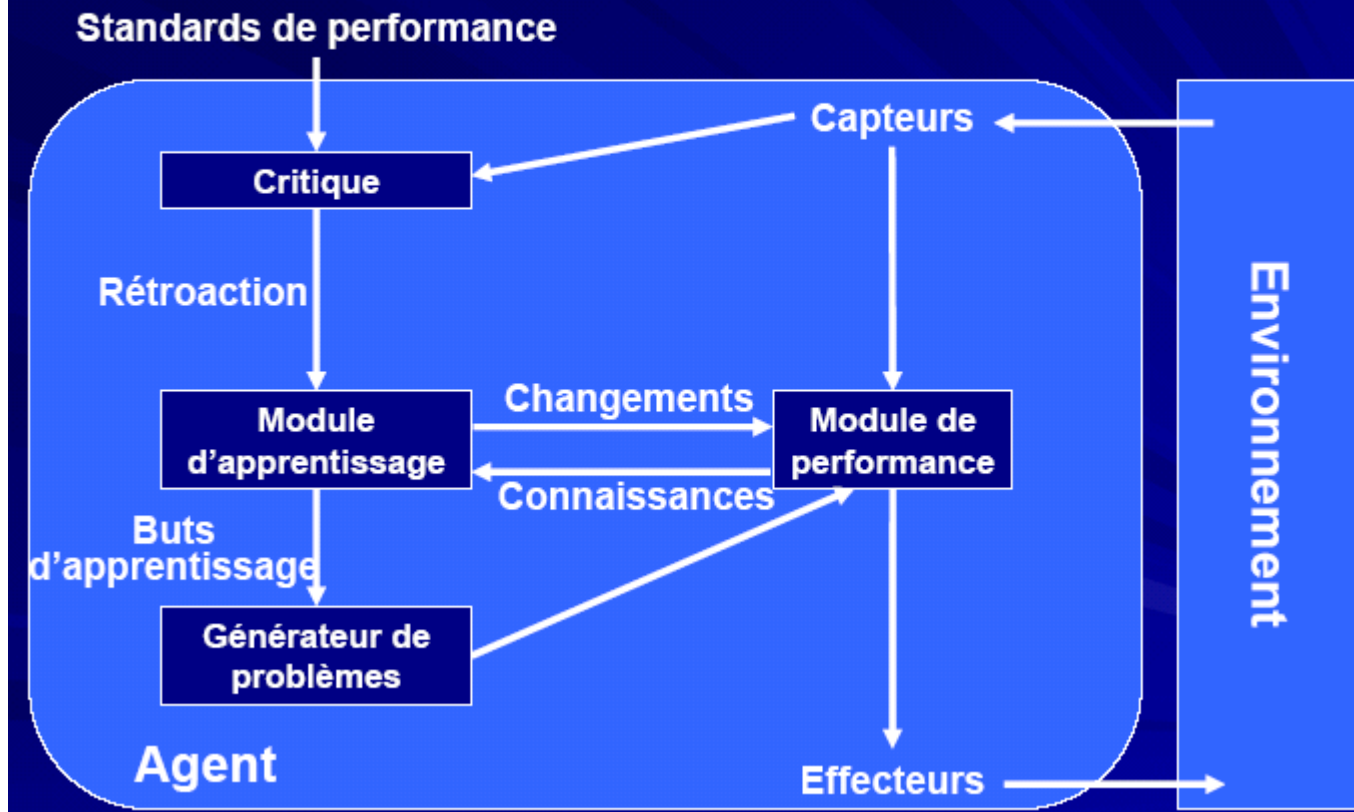
## Agent apprenant

Souvent, il est très fastidieux où même impossible de définir le comportement de l'agent à la conception.

L'apprentissage permet:

- De simplifier la conception
- À l'agent d'avoir plus de flexibilité
- À l'agent d'agir dans des environnements inconnus et de devenir meilleur avec le temps.

# Agent apprenant



# Agent apprenant

## Module de performance

- Connaissances et procédures pour choisir les actions.

## Critique

- Observe l'agent et donne des informations au module d'apprentissage.

## Module d'apprentissage

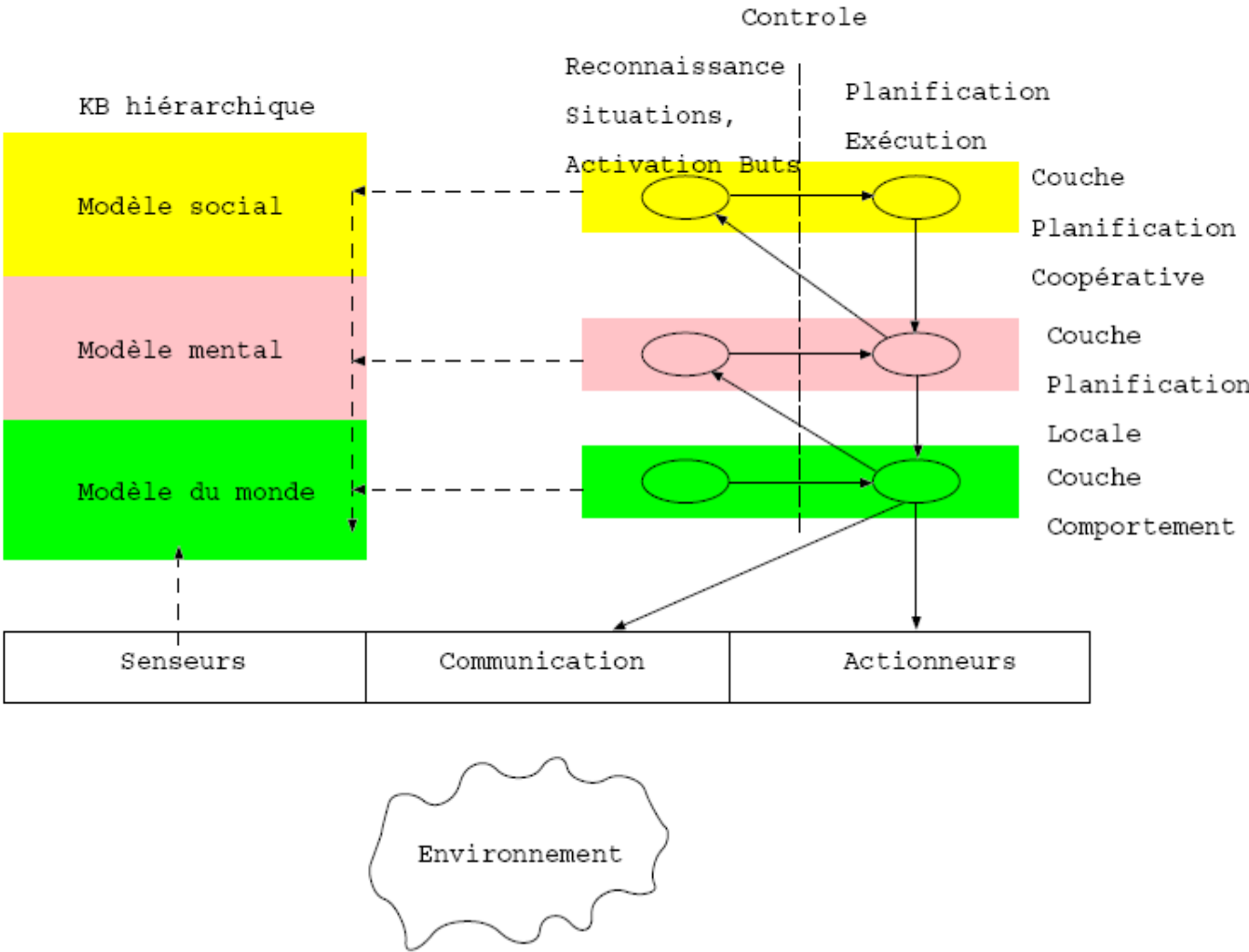
- Modifie le module de performance.

## Générateur de problèmes

- Identifie les possibilités d'amélioration et suggère des expérimentations.



Architecture hiérarchique d'un agent



## Propriétés de l'environnement

- **Complètement observable vs partiellement observable:** Est-ce que les capteurs de l'agent lui donne accès à l'état complet de l'environnement à tout moment ?
- **Déterministe vs stochastique:** Est-ce que le prochain état de l'environnement est complètement déterminé par son état courant et l'action de l'agent ?
- **Épisodique vs séquentielle:** Un épisode est une séquence perception-action. Le prochain épisode ne dépend pas des actions effectuées dans les épisodes précédents.
- **Statique vs dynamique:** Est-ce que ce que l'environnement change pendant que l'agent agent délibère ?

## Propriétés de l'environnement

- **Discret vs continu:** La distinction entre discret et continu peut être appliquée à l'état de l'environnement, à la façon dont le temps est géré, et aux perceptions et aux actions de l'agent.
- **Un agent vs multi-agent:** Est-ce qu'il y a plus qu'un agent interagissant ensemble ?

# Propriétés de l'environnement

Situation la plus difficile:

- Partiellement observable
- Stochastique
- Séquentielle
- Dynamique
- Continue
- Multi-agent

Exemple: conduite automatisée d'un taxi

# Exemples

| Environnement             | Observable    | Déterministe | Épisodique | Statique  | Discret | Agents |
|---------------------------|---------------|--------------|------------|-----------|---------|--------|
| Mots-croisés              | Complètement  | Déterministe | Séquentiel | Statique  | Discret | Un     |
| Échec avec une horloge    | Complètement  | Stratégique  | Séquentiel | Semi      | Discret | Multi  |
| Poker                     | Partiellement | Stratégique  | Séquentiel | Statique  | Discret | Multi  |
| Backgammon                | Complètement  | Stochastique | Séquentiel | Statique  | Discret | Multi  |
| Conduire un taxi          | Partiellement | Stochastique | Séquentiel | Dynamique | Continu | Multi  |
| Diagnostic médical        | Partiellement | Stochastique | Séquentiel | Dynamique | Continu | Un     |
| Analyse d'image           | Complètement  | Déterministe | Épisodique | Semi      | Continu | Un     |
| Robot ramasseur de pièces | Partiellement | Stochastique | Épisodique | Dynamique | Continu | Un     |
| Contrôleur de raffinerie  | Partiellement | Stochastique | Séquentiel | Dynamique | Continu | Un     |
| Enseignant interactif     | Partiellement | Stochastique | Séquentiel | Dynamique | Discret | Multi  |

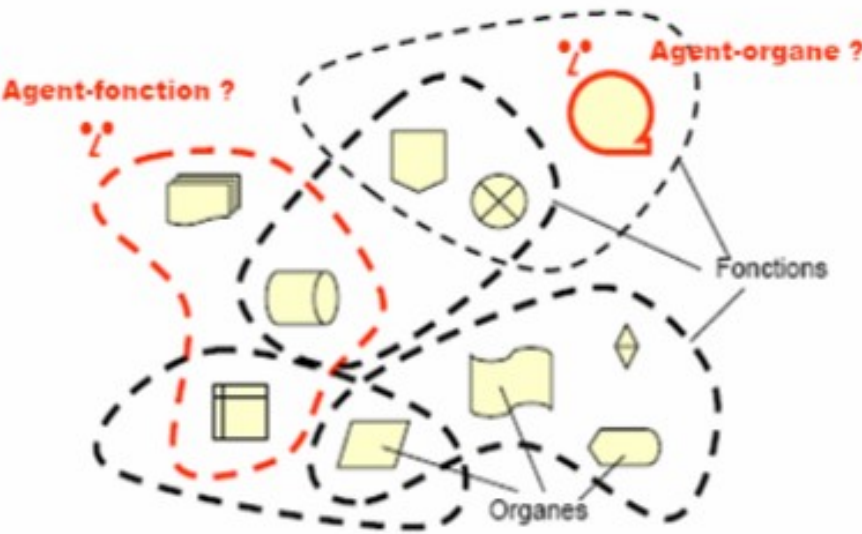
# Méthodologies de Conception

## Analyse fonctionnelle

L'analyse fonctionnelle d'une organisation connue a priori conduit à l'identification des principales fonctions que les composants (entités physiques) doivent remplir.

Exemples de fonctions :

- Fonction représentationnelle
- Fonction de contrôle (scheduling de tâches)
- Fonction conative : gestion de l'action vers l'extérieur (buts, contrats)
- Fonction communicationnelle
- Fonction opératoire : ce que l'agent rend comme service
- Fonction végétative : métabolisme, intendance.

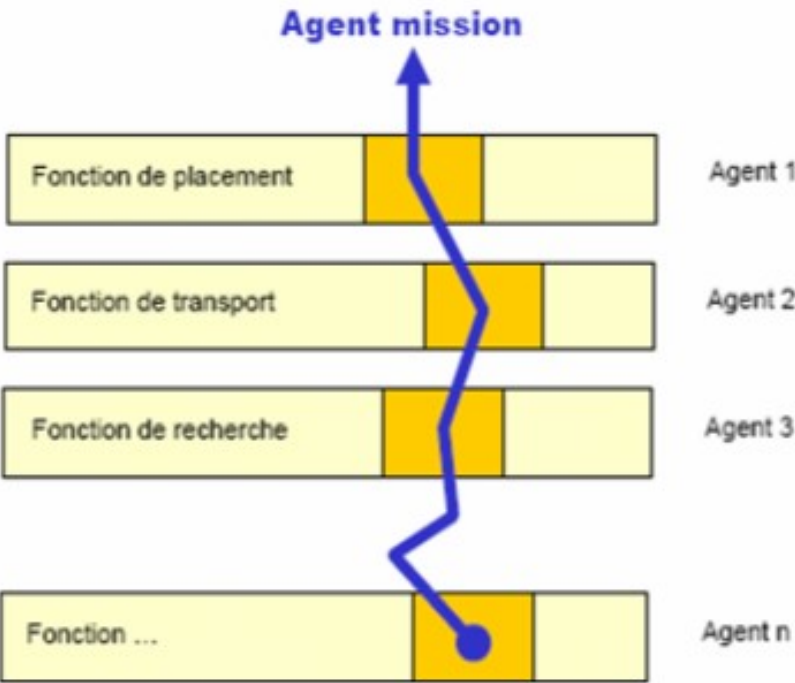


La problématique de la relation fonction ⇔ organe

## Analyse structurale

L'analyse structurale d'une organisation correspond au problème classique de l'organisation des entreprises :

- Structure horizontale : orienté savoir-faire (spécialisation des tâches),
- Structure verticale : orienté projet (but, mission)



# CONCLUSION

- Un agent perçoit et agit dans un environnement. Il a une architecture et est réalisé par un programme.
- Idéalement un agent choisit toujours l'action qui optimise sa performance attendue (étant donné la séquence perceptive reçue).
- Un agent fait correspondre une action à la séquence perceptive reçue et met à jour son état interne.
- Les agents reflex répondent immédiatement aux séquences perceptives.
- Les agents avec objectif agissent pour réaliser leur propre objectif.
- Les agents avec utilité optimisent leur propre fonction d'utilité.
- Certains environnements sont plus difficiles que d'autres; les plus difficiles sont inaccessibles, non-déterministes, non-épisodiques, dynamiques et continus, un seul agent ou multi-agents.



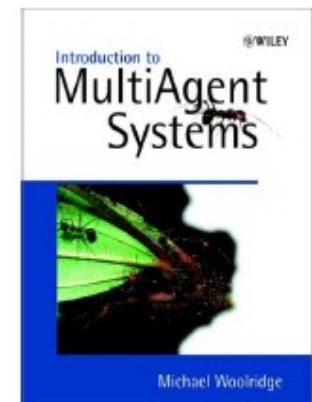
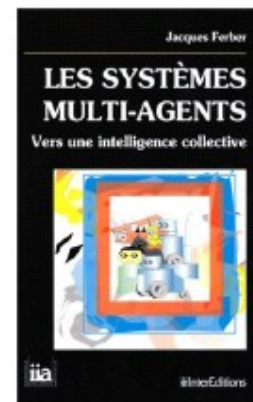
# Références

## En français

- Jacques Ferber, *Les systèmes multi agents*, InterEditions 1995 réédité en 200. La référence en français.
- Eric Bonabeau et Guy Theraulaz, *Intelligence Collective*, Hermès 1994. Tout sur la « Swarm Intelligence ».
- JFIADSMA, tous les Actes des *Journées Francophones en Intelligence Artificielle Distribuée et Systèmes Multi Agents*, Colloque annuel.

## En anglais

- Stuart Russel, Peter Norvig, *Artificial Intelligence — a modern approach*, la version 2002 s'intitule « the agent release » et intègre les concepts agents dans le cadre de l'IA classique.
- Alan H. Bond, Les Gasser Ed., *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, 1988. Un recueil très classique des articles fondateurs sur l'IAD.
- Michael J. Wooldridge, *An Introduction to multi agent systems*, J. Wiley & sons, 2002. Une bonne introduction aux agents rationnels, fondés sur les logiques (LPO et modales).





# Travaux pratiques d'IA

## Problème No 1

Pour chacun des agents suivants, développer une description PAGE.

- Un robot footballeur.
- Un agent internet de vente de livre (amazon).
- Un véhicule autonome d'exploration de mars.
- Un système de preuve de théorèmes mathématique.

## Problème No 2

Pour tout les agents listé au problème No 1 caractériser l'environnement à partir des caractéristiques vues au cours.

## Problème No 3

Considérer l'environnement du problème de l'aspirateur dans lequel l'agent est pénalisé d'un point pour chaque déplacement. .

1. Est ce qu'un agent réflexe simple peut agir d'une façon parfaitement rationnelle dans cet environnement? expliquer.
2. Que peut on dire d'un agent réflexe avec états.
3. Comment répondriez vous au deux questions précédentes si le problème change de façon à ce que la perception de l'agent nous indique l'état propre/sale de chacune des cases de l'environnement.