

# Les architectures des agents

---



# *INTRODUCTION*

**Vision fonctionnelle** : degré d'intégration des facettes A, E, I et O dans l'agent

➤ *autonomie / interagissant / social*

**Vision décisionnelle** : degré de couplage à l'environnement

➤ *réactif / hybride / délibératif*

# Architecture des SMA

---

Les architectures d'agents sont regroupées en trois classes:

- ❑ **Agent réactif** {
  - Les agents à réflexes simples
  - Les agents conservant une trace du monde
  
- ❑ **Agent délibératif** (cognitif) {
  - Les agents ayant des buts
  - Les agents utilisant une fonction d'utilité
  - Les agents BDI (*Belief, Desire, Intentions*)
  
- ❑ **Agent hybride**

# SMA: Formalisme

---

$S = \{s_1, s_2, \dots\}$  l'ensemble des états de l'environnement.

$A = \{a_1, a_2, \dots\}$  l'ensemble des actions de l'agent

$P = \{p_1, p_2, \dots\}$  l'ensemble des perceptions de l'agent

On peut modéliser un environnement non-déterministe par la fonction

$$env : S \times A \rightarrow \wp(S)$$

$\wp(S)$  ensemble d'états possibles pour l'environnement.



# SMA: Formalisme

---

Un historique  $h$  est alors défini par la séquence suivante :

$$h : s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots \xrightarrow{a_{u-1}} s_u \xrightarrow{a_u} \dots$$

où  $s_0$  est l'état initial de l'environnement,  $a_u$  est la  $u^{\text{ième}}$  action choisie par l'agent et  $s_u$  et le  $u^{\text{ième}}$  état de l'environnement (qui est l'un des résultats possibles de l'action  $a_u$  dans l'état  $s_{u-1}$ ).

# SMA: Formalisme

---

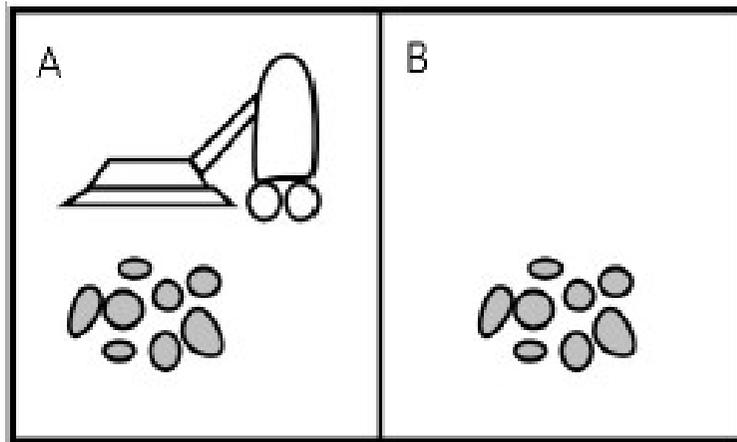
Agent  $N = (P; A; f)$  où

$f: P^* \rightarrow A$  est la FONCTION D'AGENT

(\*) : séquence

Le programme d'agent est exécuté sur une architecture physique pour réaliser (calculer)  $f$

# SMA: Formalisme



$AS = (P; A; f)$  où  
 $P = \text{Loc} \times \text{Etat}$  où  
 $\text{Loc} = \{A, B\}$  et  
 $\text{Etat} = \{ \text{propre}, \text{sale} \}$   
 $P$  est un ensemble de couples  
 (Location, Etat de propriété),  
 par ex. (A, propre)  
 $A = \{ \text{gauche}, \text{droite}, \text{aspire} \}$

Séquence de perceptions	Action
(A, propre)	<i>droite</i>
(A, sale)	<i>aspire</i>
(B, propre)	<i>gauche</i>
(B, sale)	<i>aspire</i>
(A, propre)(A, propre)	<i>droite</i>
(A, propre)(A, sale)	<i>aspire</i>
⋮	⋮

## Mesures de performance

- ✓ Quantité de poussière aspirée
- ✓ Surface nettoyée
- ✓ Consommation d'énergie
- ✓ Niveau de bruit de l'appareil
- ✓ ...

# SMA: Agent Réactif

---

- ✘ Pas de connaissances explicites ni de l'environnement, ni des autres agents, ni du passé,
- ✘ Pas de plan d'actions
- ✘ Comportement simple de type réflexe qualifié aussi de biologique
- ✘ Nombre assez important

→ Emergence

# SMA: Agent Réactif

---

$$env : \mathbf{S} \times \mathbf{A} \rightarrow \wp(\mathbf{S})$$

$$agir : \mathbf{S}^* \rightarrow \mathbf{A}$$

Intuitivement l'agent choisit l'action suivante en analysant les différents états de l'environnement

# SMA: Agent Réactif: Agent à reflexe simple

---

- **Les agents à réflexes simples**
- **Les agents conservant une trace du monde**

# SMA: Agent Réactif

## Agent à reflexe simple

---

Les agents réactifs agissent sans tenir compte de leur passé (historique). Ils réagissent uniquement à l'environnement ; leur comportement est défini par la fonction :

$agir : S \rightarrow A$  .

C'est le cas de l'agent *thermostat* dont le comportement est le suivant :

$$agir(s) = \begin{cases} augmenter\ OFF\ si\ s =\ température\ OK \\ augmenter\ ON\ sinon \end{cases}$$

agissent sans

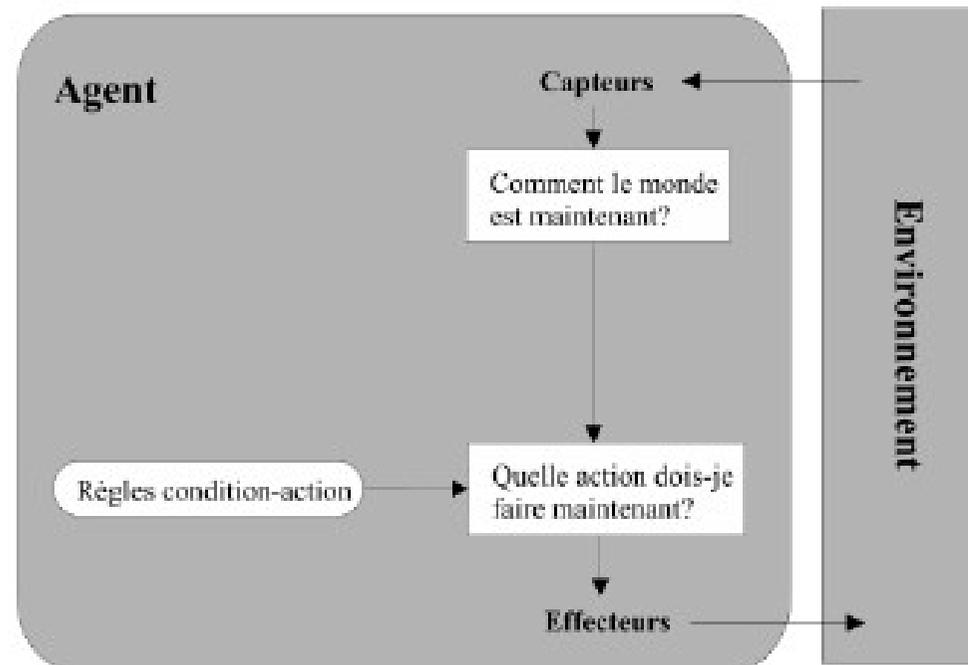
tenir compte du passé (historique), ni du futur (plan

d'actions).



# SMA: Agent Réactif: Agent à reflexe simple

---



*capter* :  $S \rightarrow P$  , qui fait correspondre les états de l'environnement à des perceptions qu'en a l'agent ; ce dernier est défini par :

*agir* :  $P^* \rightarrow A$  , qui fait correspondre une séquence (\*) de perceptions à des actions.

# SMA: Agent réactif: Agent à reflexe simple

---

*règles* : règles condition-action

*perceptions* : ensemble des perceptions ( $P^*$ )

**répéter**

*état* := interpréter\_entrée (*perception*) ;

*règle* := match (*état*, *règles*) ;

agir (*règle* [*action*] ) ;

**indéfiniment**

# SMA: Agent réactif: Agent à reflexe simple

---

Programme pour l'aspirateur

```
function AGENT-ASPIRATEUR( lieu, statut) returns act  
  
  if status = sale then return aspire  
  else if endroit = A then return droite  
  else if endroit = B then return gauche
```

# SMA: Agent réactif: Agent à reflexe simple

---

D'où des propriétés intéressantes comme :

$$s_1 \in S \wedge s_2 \in S, \text{ avec } s_1 \neq s_2 \text{ mais } \text{capter}(s_1) = \text{capter}(s_2)$$

où l'agent ne fait pas la distinction entre les deux états de l'environnement, qui lui sont devenus indiscernables.

# SMA: Agent réactif: Agent à reflexe simple

---

## *Les agents à réflexes simples*

Soit les deux faits suivants de l'environnement :

$x$  : "La température de la pièce est bonne".

$y$  : "Le chat est dans la pièce".

L'ensemble des 4 états ( $s_1, s_2, s_3, s_4$ ) possibles est :

$$S = \{ \{ \neg x, \neg y \}, \{ \neg x, y \}, \{ x, \neg y \}, \{ x, y \} \}$$

Les perceptions  $p_1$  et  $p_2$  du thermostat sont :

$$\text{capter}(s) = \begin{cases} p_1 \text{ si } s = s_1 \text{ ou } s = s_2 \\ p_2 \text{ si } s = s_3 \text{ ou } s = s_4 \end{cases} \quad x \wedge \neg x, \quad \forall y$$

# SMA: Agent réactif: Agent à reflexe simple

---

En effet le thermostat ne perçoit pas le fait "y".

Pour lui :

capter ( $s_1$ ) = capter ( $s_2$ ), et

capter ( $s_3$ ) = capter ( $s_4$ ).

Les états ( $s_1, s_2$ ) et ( $s_3, s_4$ ) sont équivalents pour lui.

# SMA: Agent réactif: Agent à reflexe simple

---

- L'agent peut choisir ses actions en se basant uniquement sur sa perception actuelle.
- Ils n'ont pas une représentation symbolique de l'environnement, des connaissances.
- L'agent peut avoir deux perceptions identiques mais qui sont en réalité différentes.
- « Les capteurs » de l'agents ne fournissent pas une vue complète sur l'état du monde.
- Manque de flexibilité



L'agent doit maintenir des informations internes sur l'état de l'environnement.

---

# Agent Cognitif (Délibératif)



# SMA: Agent Cognitif

---

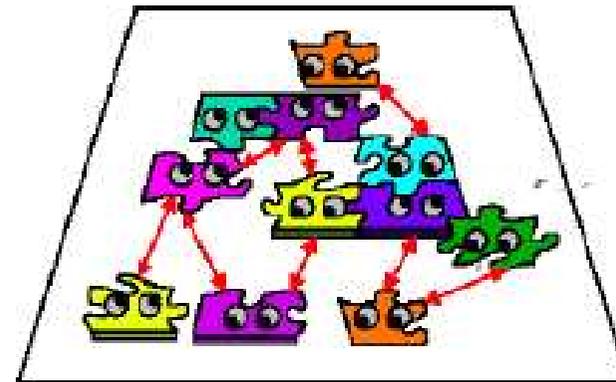
- Volonté de faire communiquer des Systèmes experts classiques
- SMA = petit nombre d'agents intelligents comprenant une base de connaissance : info + des savoirs faire
- L'agent comprend un modèle du monde représenté explicitement sous la forme d'un modèle symbolique
- Le module de décision de tels agents est réalisé via le raisonnement logique.

# SMA: Agent Cognitif

---

## □ Agents réactifs

- agents = parties du problème lui-même
- décomposition structurelle
- approche ascendante
- comportement collectif émergent



## □ Agents cognitifs (délibératifs)

- agents = experts, solveurs
- multi-expert
- décomposition fonctionnelle
- approche descendante
- coordination explicite



# SMA: Agent Cognitif

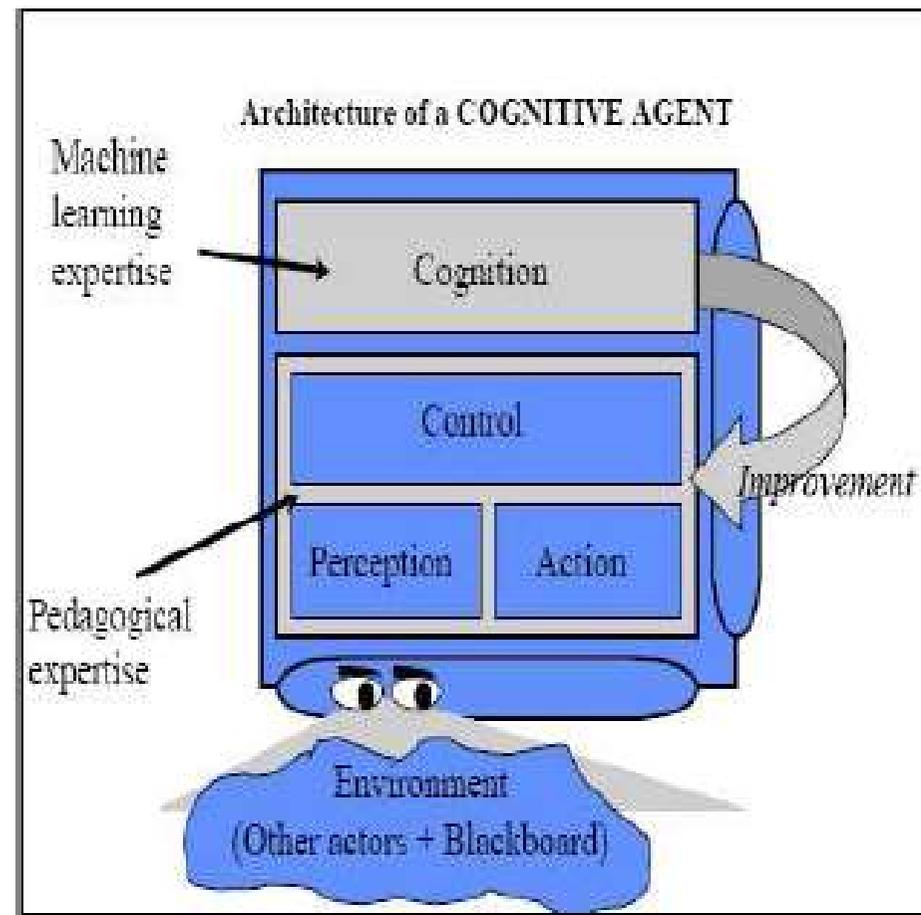
---

- ✦ Connaissances explicites partielles de l'environnement, des autres agents, du passé
- ✦ Plan d'actions
- ✦ Comportement intelligent qualifié de social
- ✦ Nombre assez réduit
- ✦ Base de connaissance assez importante
- ✦ Coopération, coordination
- ✦ Des explications

# SMA: Agent Cognitif

Deux processus

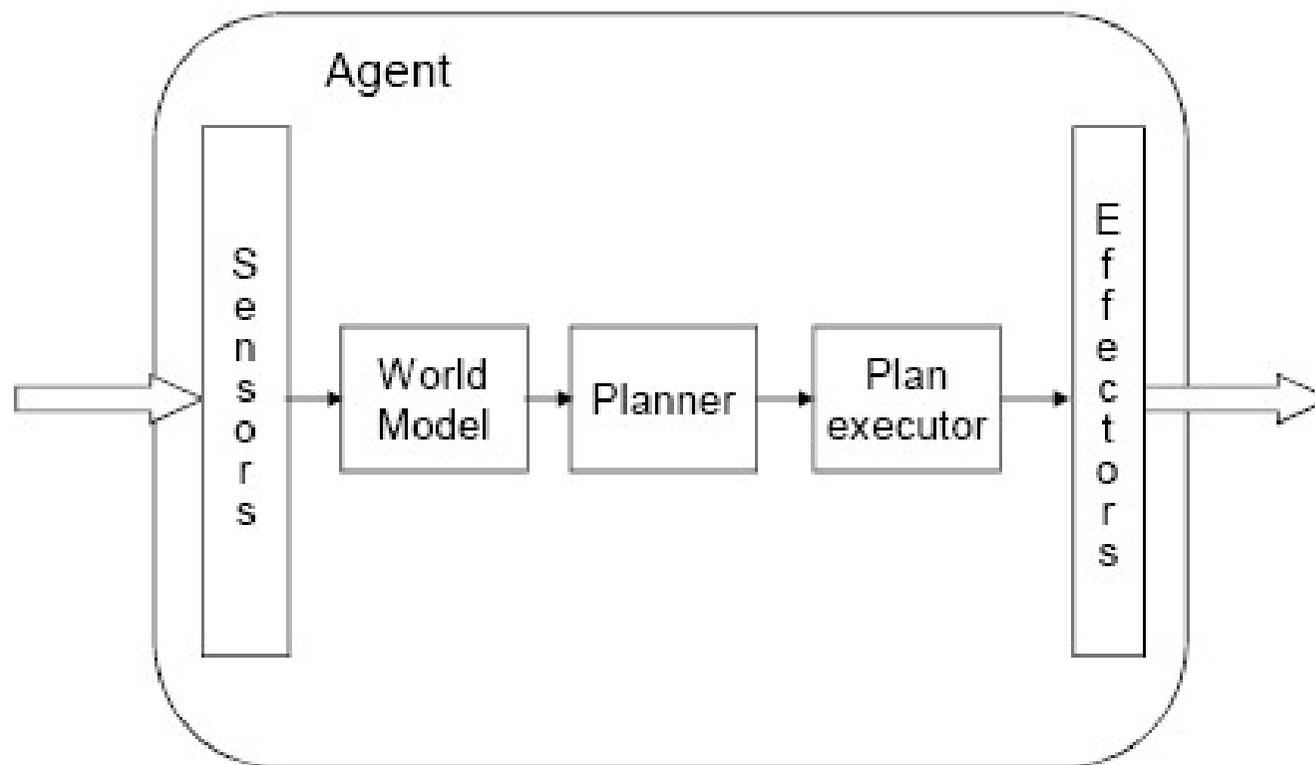
- Décider quels buts poursuivre : **Quoi ?**  
⇒ délibérations
- Décider comment les réaliser : **Comment ?**  
⇒ *means-end reasoning*



# SMA: Agent Cognitif

---

## Notion de plan



# SMA: Agent Cognitif

---

## Notion de plan

Soit  $E$  la représentation du monde actuel

- Et soit  $G$  la représentation du monde qu'on veut atteindre
- Un problème de planification noté  $(E;G)$  consiste à trouver la suite d'actions  $a_1; \dots; a_n$

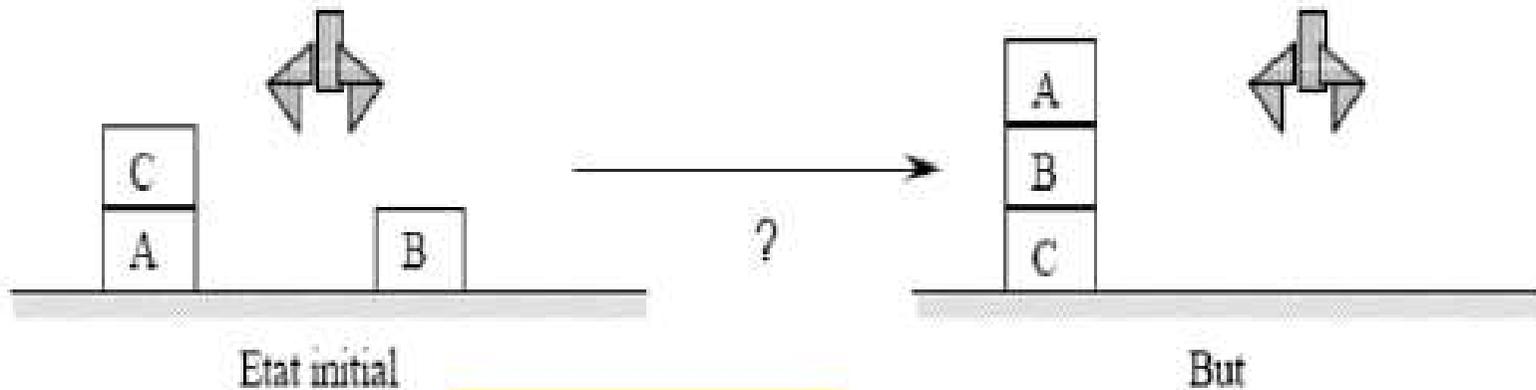
Tels que :



Un plan est une suite d'actions qui appliquées sur l'état initial  $E$  mène au but  $G$ .

# SMA: Agent Cognitif

## Notion de plan



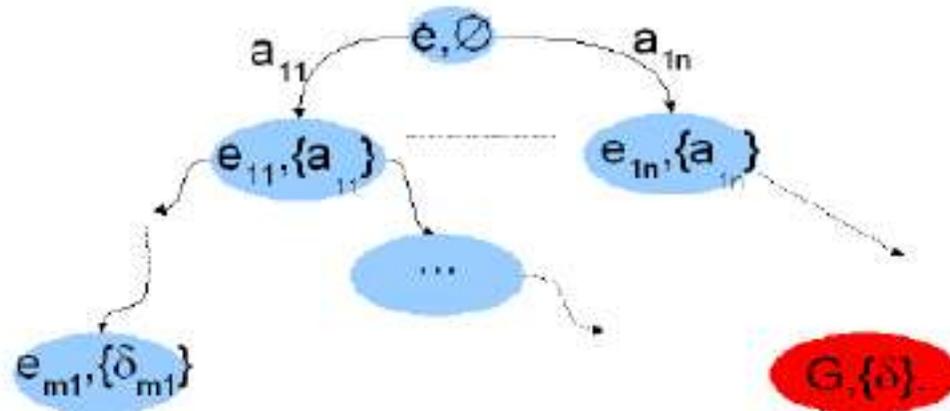
```
pickup(C) ;  
putdown(C) ;  
pickup(B) ;  
stack(B,C) ;  
pickup(A) ;  
stack(A,B)
```

# SMA: Agent Cognitif

---

## Notion de plan

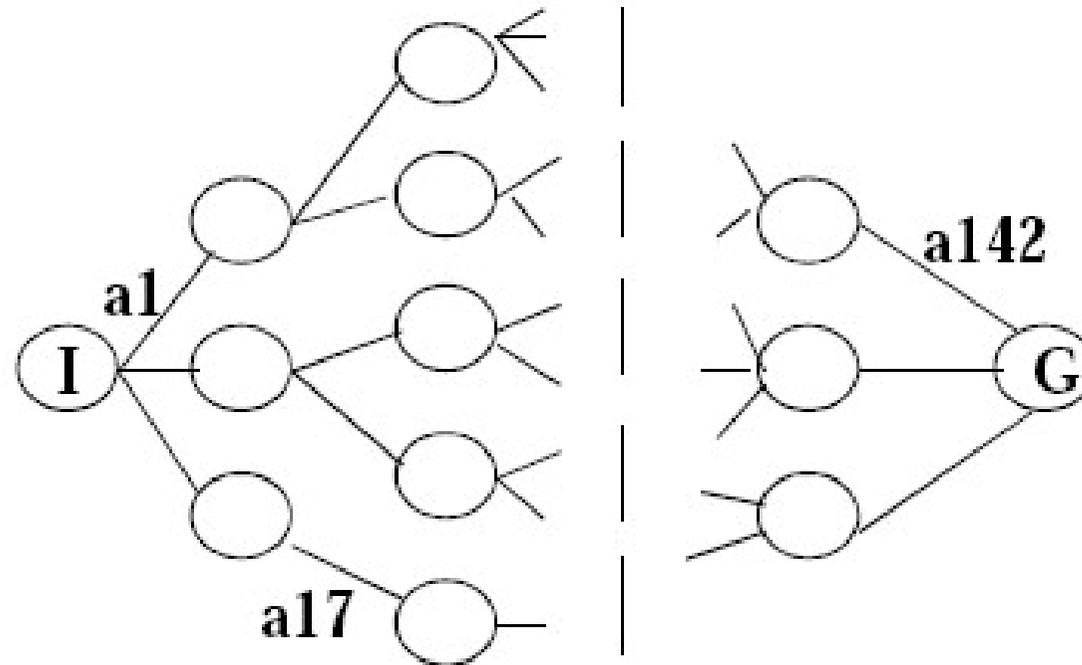
- Étant donné un état  $E$
- Nous pouvons construire un graphe d'états où chaque état est un couple : Etat du monde et liste des actions qui mène de l'état du monde initial vers l'état en question.



# SMA: Agent Cognitif

---

## Notion de plan



**Un plan est une séquence d'actions**

# SMA: Agent Cognitif

---

Effectue une certaine délibération pour choisir son action. Une telle délibération peut se faire en se basant sur:

- les buts de l'agent ou,
- sur une certaine fonction d'utilité.

# SMA: Agent Cognitif

---

- **Les agents ayant des buts**
- **Les agents utilisant une fonction d'utilité**
- **Les agents BDI (*Belief, Desire, Intentions*)**

# SMA: Agent Cognitif

---

Plusieurs type d'agents cognitifs:

- Agents ayants un but
- Agents utilisant une fonction d'utilité
- Agent BDI (Belive, Desire, Intention)

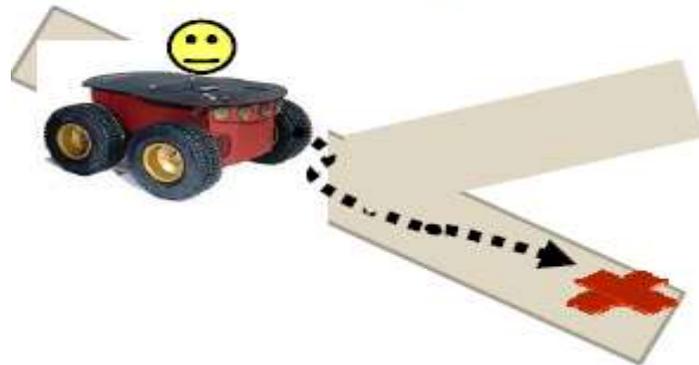
# SMA: Agent Cognitif

---

Plusieurs type d'agents cognitifs:

- Agents ayants un but

## *Les agents ayant un but*



*Exemple: Un taxi automatique arrive à une intersection. S'il ne sait pas quelle est sa destination finale, il ne saura pas choisir parmi les différentes routes.*

- Agents utilisant une fonction d'utilité
- Agent BDI (Belive, Desire, Intention)

# SMA: Agent Cognitif BDI

---

Plusieurs type d'agents cognitifs:

- Agents utilisant une fonction d'utilité

*Exemple:* il existe plusieurs chemins pour arriver à destination, et certains sont plus courts que d'autres.

Pour palier ce problème, il faut introduire  
*une fonction d'utilité.*



# SMA: Agent Cognitif BDI

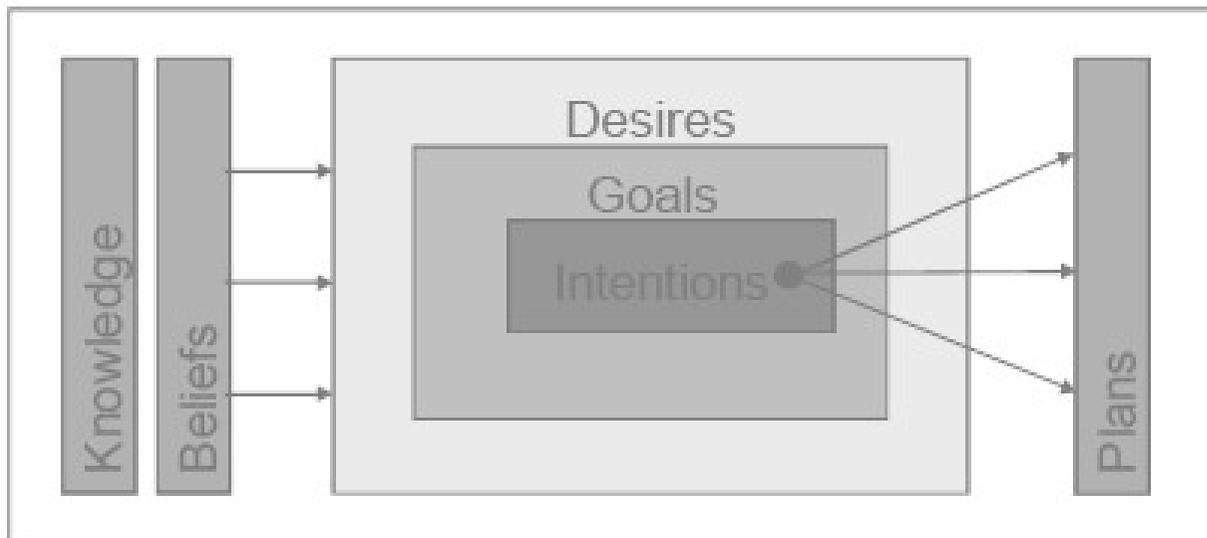
---

- **Connaissances** → Ali connaît le fait que les humains sont mortels
- **Croyances** → Ali a pris son parapluie parce qu'il croit qu'il va pleuvoir
- **Désirs, buts** → Ali désire avoir son doctorat
- **Intentions** → Ali a l'intention de travailler dur pour avoir sa thèse
- **Choix, décisions** → Ali a décidé de faire une thèse
- **Engagements** → Ali ne va pas s'arrêter de travailler avant d'avoir fini sa thèse
- **Conventions** → si, par hasard, Ali décide d'abandonner sa thèse, il va le dire à son professeur
- **Obligations** → Ali doit travailler pour entretenir sa famille

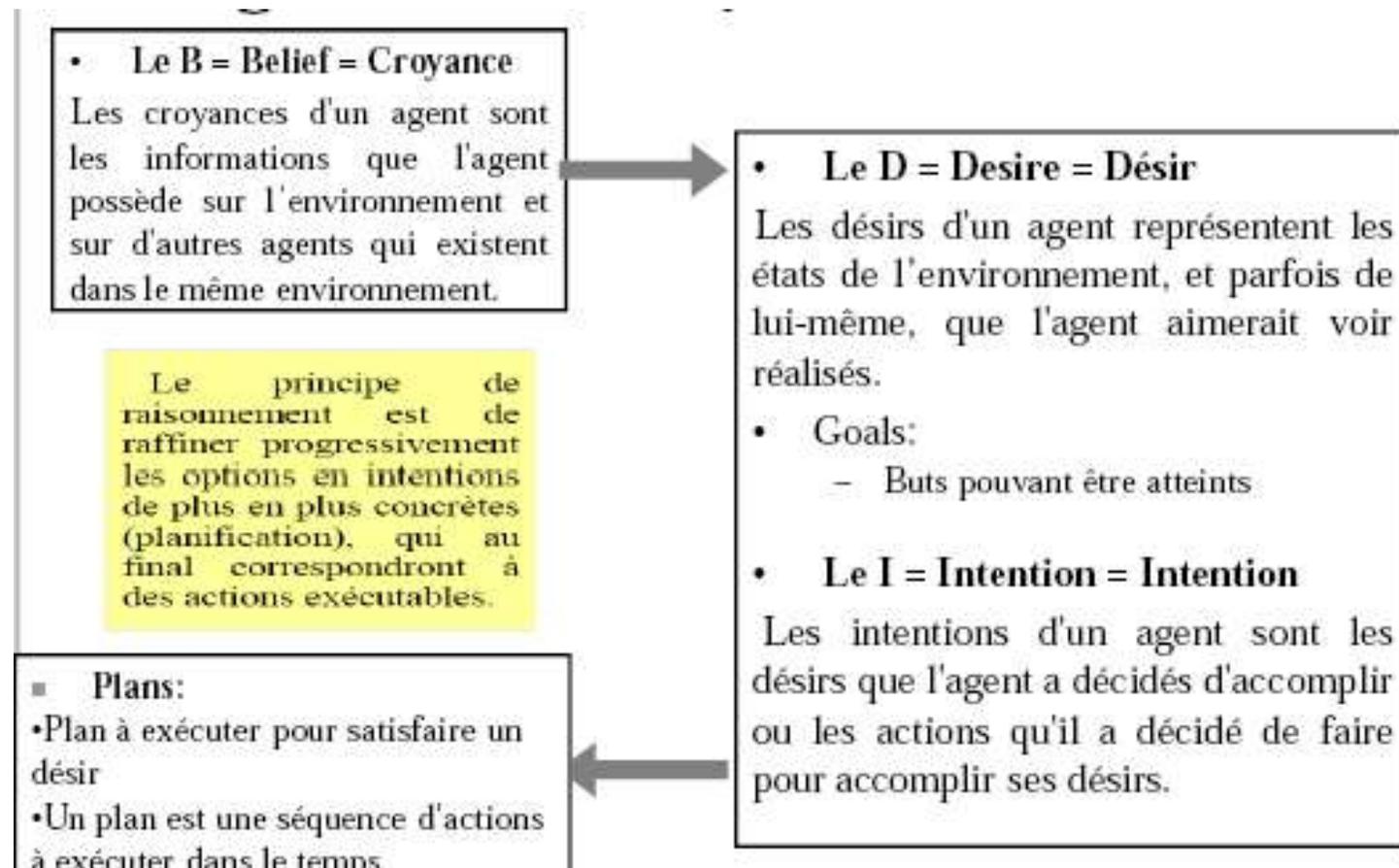
# SMA: Agent Cognitif BDI

---

- *Croyances (Belief)* : avoir certaines informations, connaissances sur le problème
- *Désires (Desire)* - : envisager les options possibles et les états que l'agent souhaite atteindre, *les états peuvent être contradictoires*
- *Intentions (Intention)* : choisir certains états à atteindre



# SMA: Agent Cognitif BDI



# SMA: Agent Cognitif BDI

---

➤ Démarche logique « Practical Reasoning »

✓ *Architecture fondée sur des extensions de la logique*

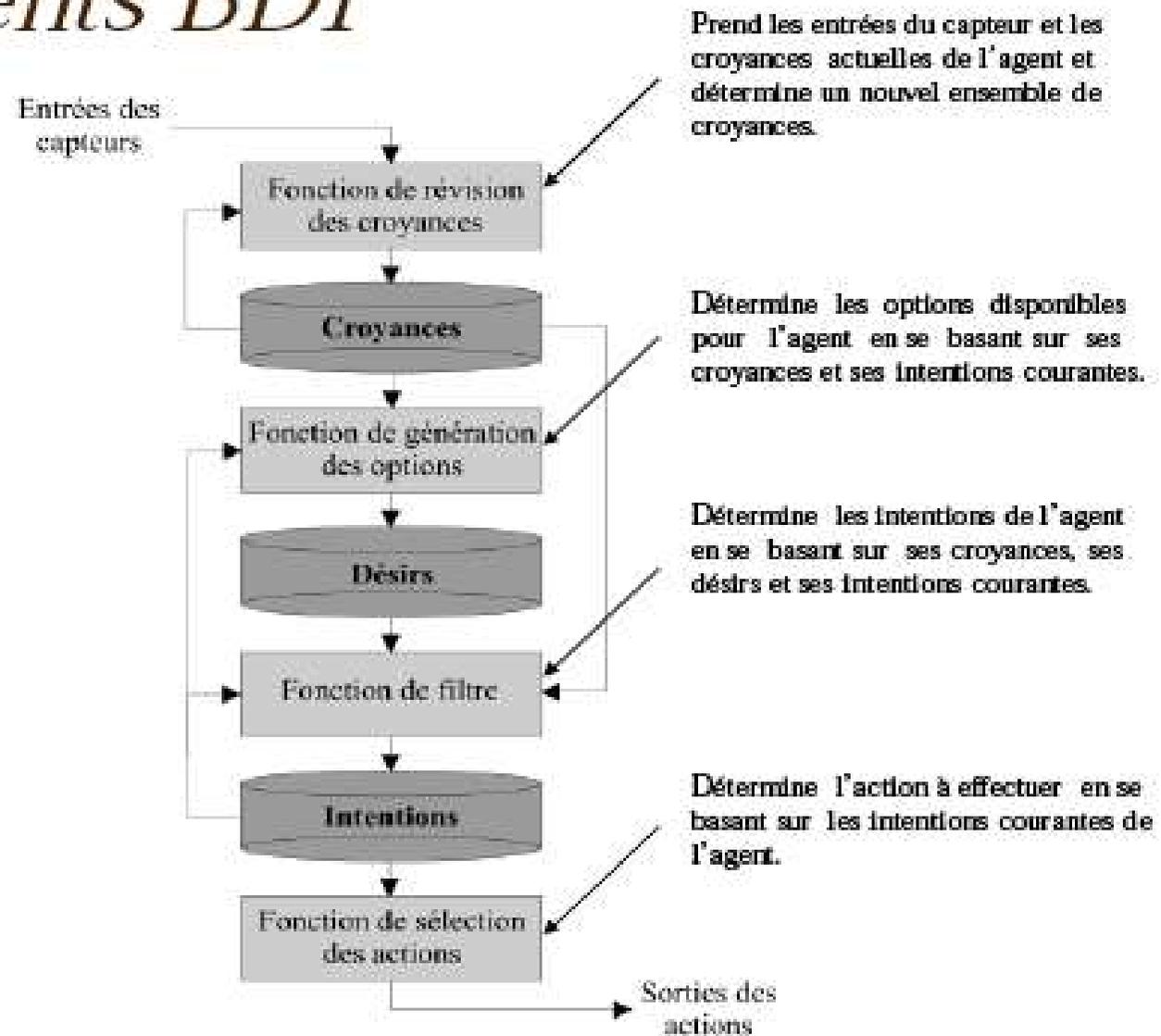
➤ Afin de prendre une décision un agent va suivre le processus suivant :

*Le processus démarre par une mise à jour  
des croyances à partir des perceptions*

1. L'agent définit ces Croyances : informations, connaissances sur l'environnement courant
2. L'agent définit ses Désirs : États que l'agent souhaite atteindre
3. L'agent définit ses Intentions : Choisir certains états à atteindre

*Le processus se termine par une fonction  
de sélection de l'action à exécuter*

# Les agents *BDI*



# SMA: Agent Cognitif BDI

---

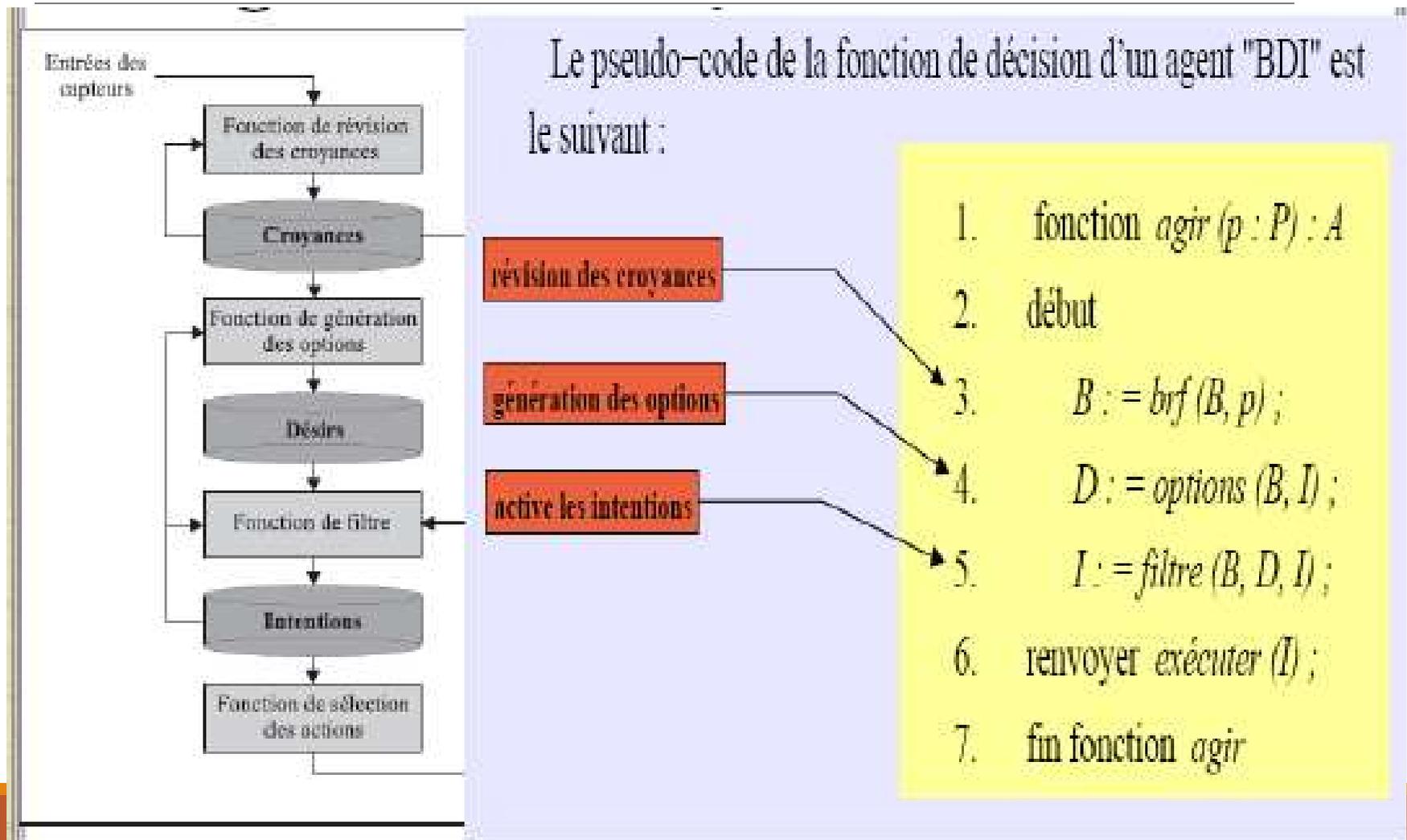
Soit :

- *Bel* est l'ensemble des **croiances** possibles,
- *Des* est l'ensemble des **désirs** (options) possibles et
- *Inten* est l'ensemble des **intentions** possibles de l'agent.

L'état de l'agent est décrit par  $(B, D, I)$  où l'on a :

$$B \subseteq Bel, D \subseteq Des, \text{ et } I \subseteq Inten$$

# SMA: Agent Cognitif BDI



# Agent Cognitif: Conclusion

---

- ❑ Ces agents disposent des capacités fondamentales de perception, délibération et action, couplés à des concepts nouveaux de représentation et de régulation des processus comportementaux.
- ❑ Les limites de cette approche sont dues à la complexité des algorithmes de manipulation symboliques

# Exercice1

---

Donner la PAGE et les caractéristiques de l'environnement d'un agent en charge de la mise en place des ouvrages dans une bibliothèque. On pourrait supposer qu'un tel agent prend les livres d'une place donnée et les range à la bonne place selon les côtes des livres.



# Exercice 2 (1)

---

Un vaisseau spatial s'est écrasé sur la lune. Afin de comprendre la cause de l'accident l'agence spatiale envoie un robot pour déterminer l'emplacement de la boîte noire de ce vaisseau.

Le robot peut se déplacer sur le champ de l'accident de façon aléatoire, pour l'aider dans sa fonction le champ de recherche est de 10000 m<sup>2</sup> s'est divisé en (100 X 100) cellules.

A chaque fois le robot trouve une chose il doit vérifier s'il s'agit vraiment de la boîte noire du vaisseau et il doit mémoriser ses coordonnées sinon il va se déplacer jusqu'à parcourir les 10000 cellules.



# Exercice 2 (2)

---

Le robot se trouve initialement à la cellule [1,1]

Find (x, y) : permet de tester si la cellule contient la boîte noire du vaisseau (true, false).

Move (x, y) : permet de se déplacer entre les cellules.

Visited (x, y) : indique que la cellule est déjà visitée par le robot (true, false).

Send (x, y) : permet d'envoyer les coordonnées à l'agence.



## Exercice 2 (3)

---

- Quelle est le type de l'agent qui correspond à ce robot ?
- Ecrire les règles qui permettent de trouver la position de la boîte noire
- Proposer l'algorithme qui permet à l'agent de trouver la boîte noire en visitant le champ indiqué.