

Agent réactif & Agent cognitif

Coordination d'Agents réactifs

La coordination d'agents réactifs s'inspire de mécanismes issus de la biologie comme les insectes sociaux :

- ❑ **communication via la modification de l'environnement physique.** Exemple : construction d'une termitière, construction collective de toile d'araignées sociales, ...
- ❑ **communication via des signaux,** par exemple en utilisant des phéromones.

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

Les actions représentant les capacités de l'agent

Le non déterminisme du comportement d'un environnement peut être modélisé par une fonction :

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

qui prend l'état courant de l'environnement s et une fonction A (accomplie par l'agent) pour donner un ensemble d'états de l'environnement $\mathbf{env}(s, a)$ qui est le résultat de l'exécution de l'action a dans l'état s .

SMA: Formalisme

- On peut représenter l'interaction de l'agent et de l'environnement comme un historique, h déterminé comme une séquence :

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

- Où s_0 est l'état initial de l'environnement a_k est la $k^{ième}$ action que l'agent a préféré exécuter et s_k est le $k^{ième}$ état de l'environnement

⊙ Agents purement réactifs(à reflexe simple)

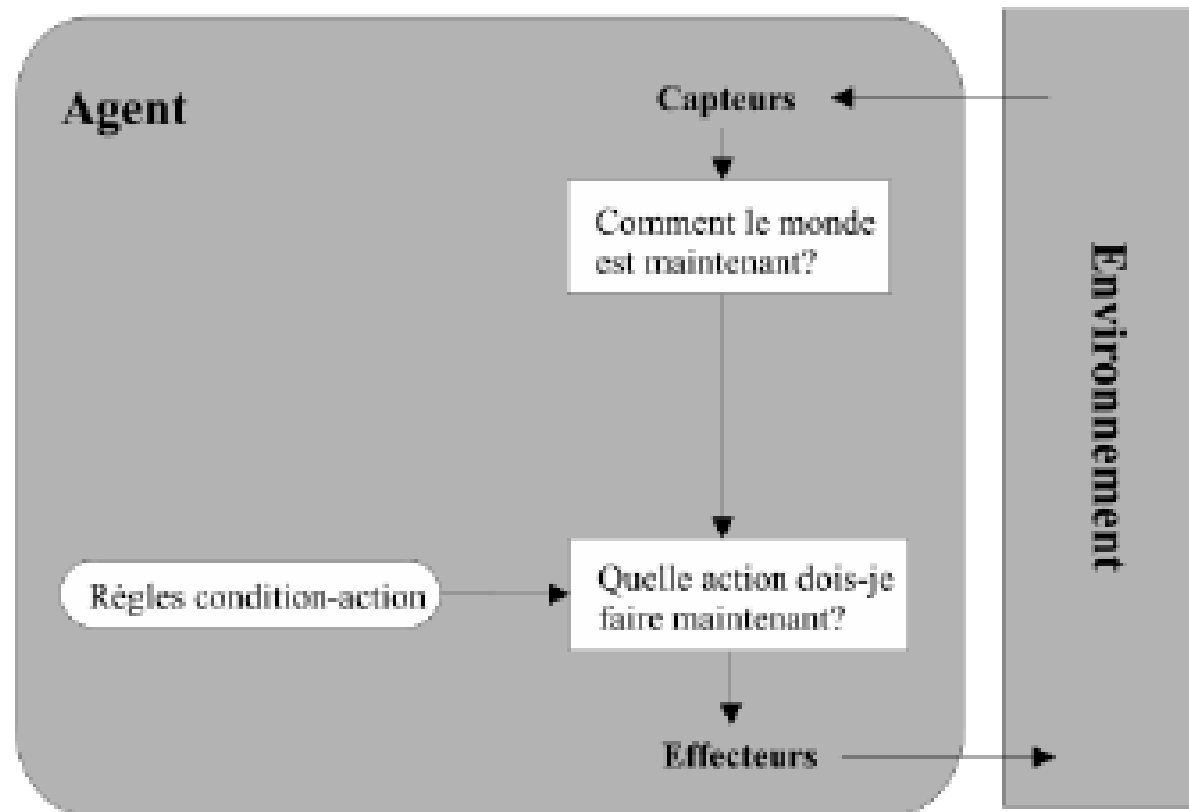
Nous allons formaliser l'architecture interne d'un agent "intelligent". Un agent peut être considéré comme une fonction

$$\text{Action: } S \rightarrow A$$

⊙ Exemple : Le thermostat

- Etats de l'environnement
 $S = \{\text{Trop froid, température OK}\}$
- $\text{Action}(S) =$
chauffage éteint **si** $s =$
température OK,
chauffage allumé **sinon**

◉ Agents purement réactifs(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.

⦿ Agents purement réactifs(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

◉ Agents purement réactifs(à reflexe simple)

Exemple : Aspirateur robotisé

Programme pour l'aspirateur

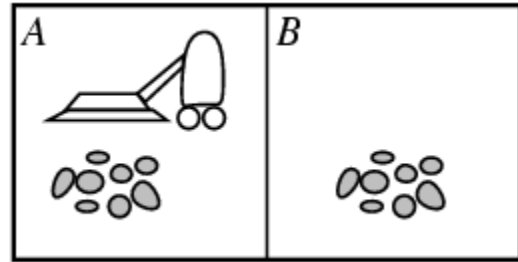
```
function AGENT-ASPIRATEUR( lieu,statut) returns act
```

```
  if status = sale then return aspire
```

```
  else if endroit = A then return droite
```

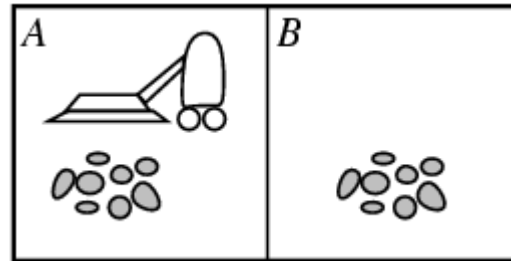
```
  else if endroit = B then return gauche
```

Exemple : Aspirateur robotisé



- Observations (données sensorielles) : position et état des lieux
Par exemple : $[A, Clean]$,
 $[A, Dirty]$,
 $[B, Clean]$,
- Actions : *Left, Right, Suck, NoOp*

Exemple : Aspirateur robotisé



- f :
 - $[A, \text{Clean}] \rightarrow \text{Right}$
 - $[A, \text{Dirty}] \rightarrow \text{Suck}$
 - ...
 - $[A, \text{Clean}] [A, \text{Clean}] [A, \text{Dirty}] \rightarrow \text{Suck}$
 - $[A, \text{Clean}] [A, \text{Clean}] [A, \text{Clean}] \rightarrow \text{Right}$
 - ...

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Architecture à subsomption (architecture interne): la plus connue des architectures d'agents réactifs.

Trois idées clés : comportement intelligent peut être généré

- (1) sans représentation explicite (à l'encontre de l'IA symbolique)
- (2) sans raisonnement abstrait explicite (à l'encontre de l'IA symbolique)
- (3) l'intelligence est une propriété émergente de certains systèmes complexes due aux **interactions**.

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Deux caractéristiques :

- ❑ La prise de décision d'un agent est réalisée à travers un ensemble de modules comportementaux correspondant à la tâche à réaliser. Comportement implémenté sous la forme de règles :

situation → action

- ❑ Pour la résolution de conflit lorsque plusieurs comportements peuvent être déclenchés à un instant donné : - arranger les modules d'action dans une hiérarchie de subsomptions selon différentes couches - les hauts niveaux peuvent inhiber les niveaux inférieurs

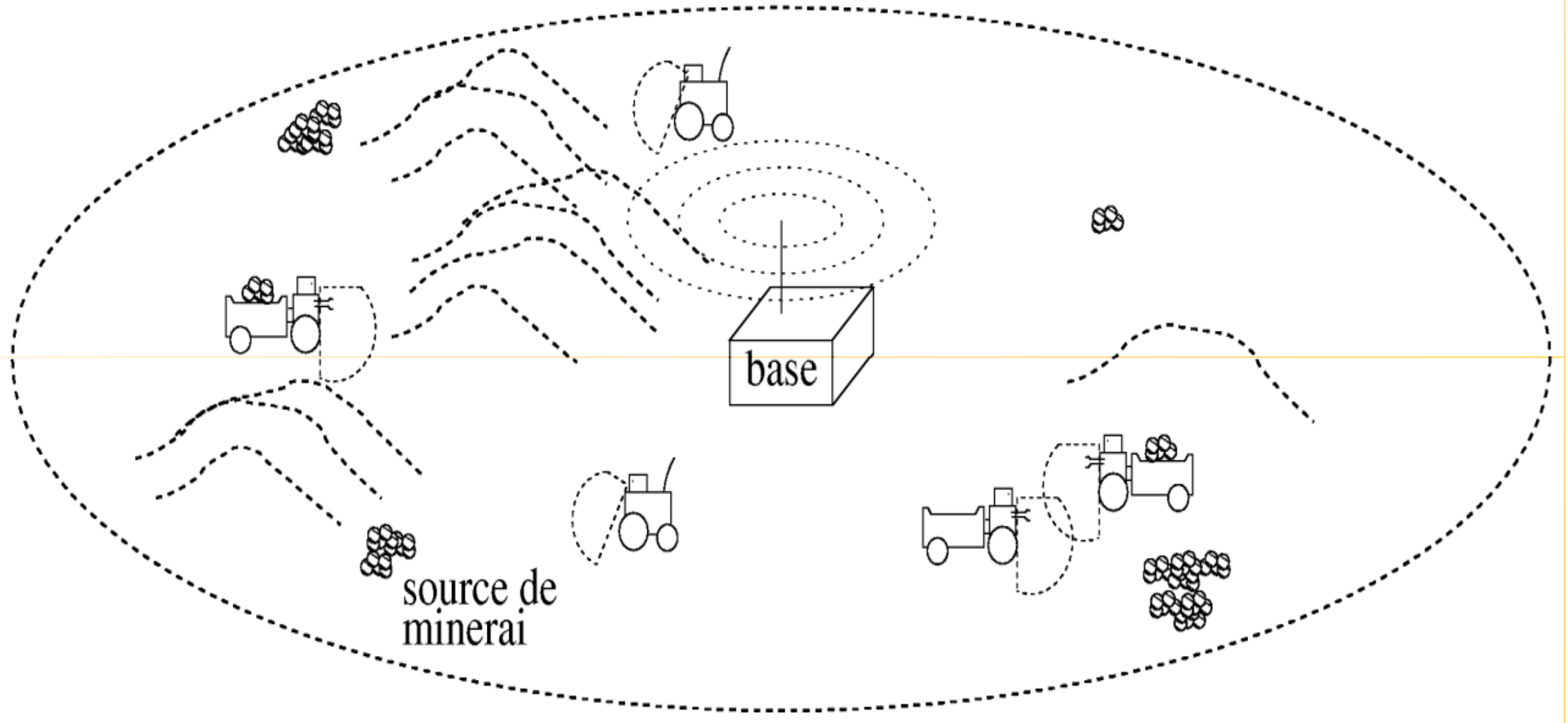
Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Exemple : les robots explorateurs

L'objectif est de réaliser un collectif de robots pour explorer une planète éloignée.

- ❑ Le but de ces robots est de collecter des échantillons de minerai sur le sol.
- ❑ La localisation de ces échantillons est inconnue au départ.
- ❑ Les robots mémorisent la localisation de la base d'où ils viennent et où ils doivent ramener les minerais collectés.
- ❑ Ils n'ont pas de carte détaillée de la région à explorer, par contre ils savent que cette région comporte des obstacles infranchissables et interdisant la communication directe.
- ❑ Pour simplifier, on suppose qu'un robot possède une énergie lui permettant de fonctionner indéfiniment.

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

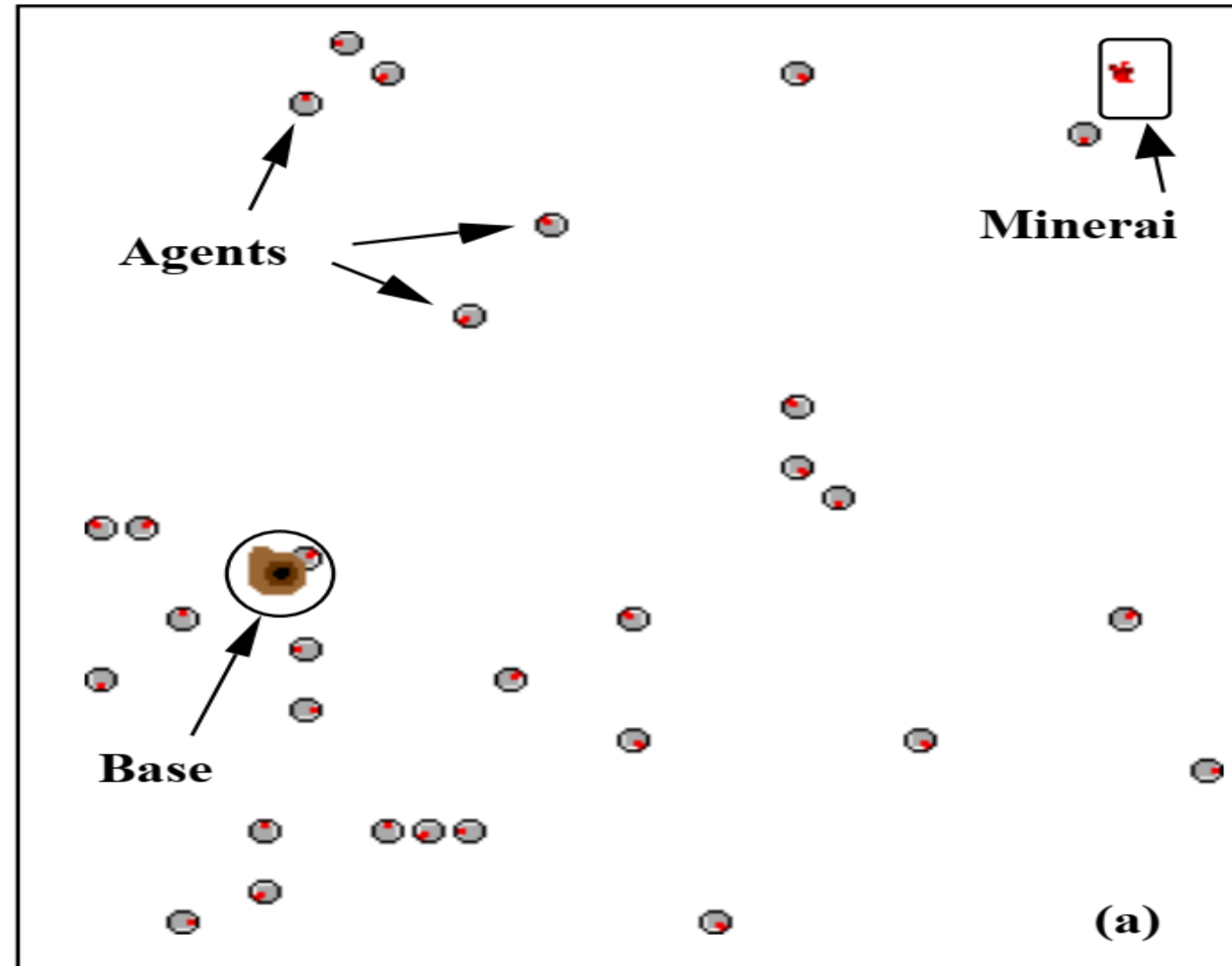


Exemple Robots explorateurs

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Exemple

Robots explorateurs



Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Ces robots doivent coopérer pour collecter les échantillons le plus efficacement possible.

Dans le cadre de l'architecture à subsomption définir :

- Les actions d'un robot
- Les règles ou modules comportementaux d'un robot 'solitaire'
- La hiérarchie de subsomptions

Même question pour un ensemble de robots coopératifs

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Modules comportementaux :

☐ Explorer

☐ Prélever

☐ Aller à la base

☐ Déposer

☐ Éviter obstacle

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Modules comportementaux pour robot sous forme de règles (non coopératif)

règle 1 : si détecte un obstacle alors changer de direction

règle 2 : si porte un échantillon et à la base alors déposer l'échantillon

règle 3 : si porte un échantillon et pas à la base alors suivre gradient (aller à la base)

règle 4 : si détecte un échantillon alors le prélever

règle 5 : si vrai alors se déplacer aléatoirement

❑ Imaginons que le robot ait le choix entre la règle 1 et la règle 3. Que doit-il faire?

❑ Même question pour les règles 4 et 3. => hiérarchie de déclenchement ?

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

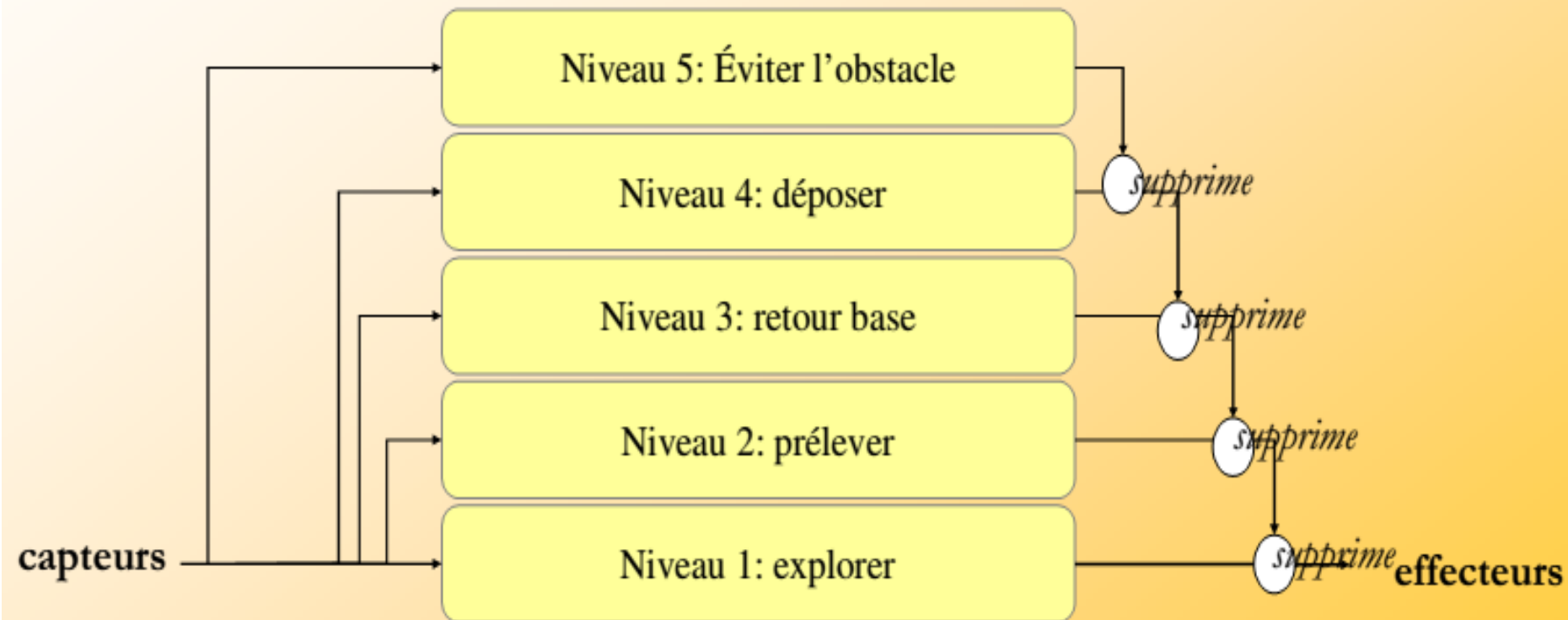
Architecture à subsomptions

Hiérarchie des priorités de déclenchement des modules d'action :

Priorité de règles: règle1 > règle2 > règle3 > règle4 > règle5

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

- Robots explorateurs de Mars (Steels 89)



Priorité Niveau 5 > priorité Niveau 4 > ...

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Exemple de règles pour robots coopératifs

Comment rendre les robots coopératifs ?

Poser deux marques « radioactives » au retour et enlever 1 marque à l'aller :

⇒ la règle 3 est remplacée par :

Règle 3bis: si porte un échantillon et pas à la base alors lâcher 2 marques radioactives et suivre gradient (retour)

⇒ une nouvelle règle :

règle 6 : si perçoit des marques alors retirer 1 marque et suivre le gradient (aller)

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

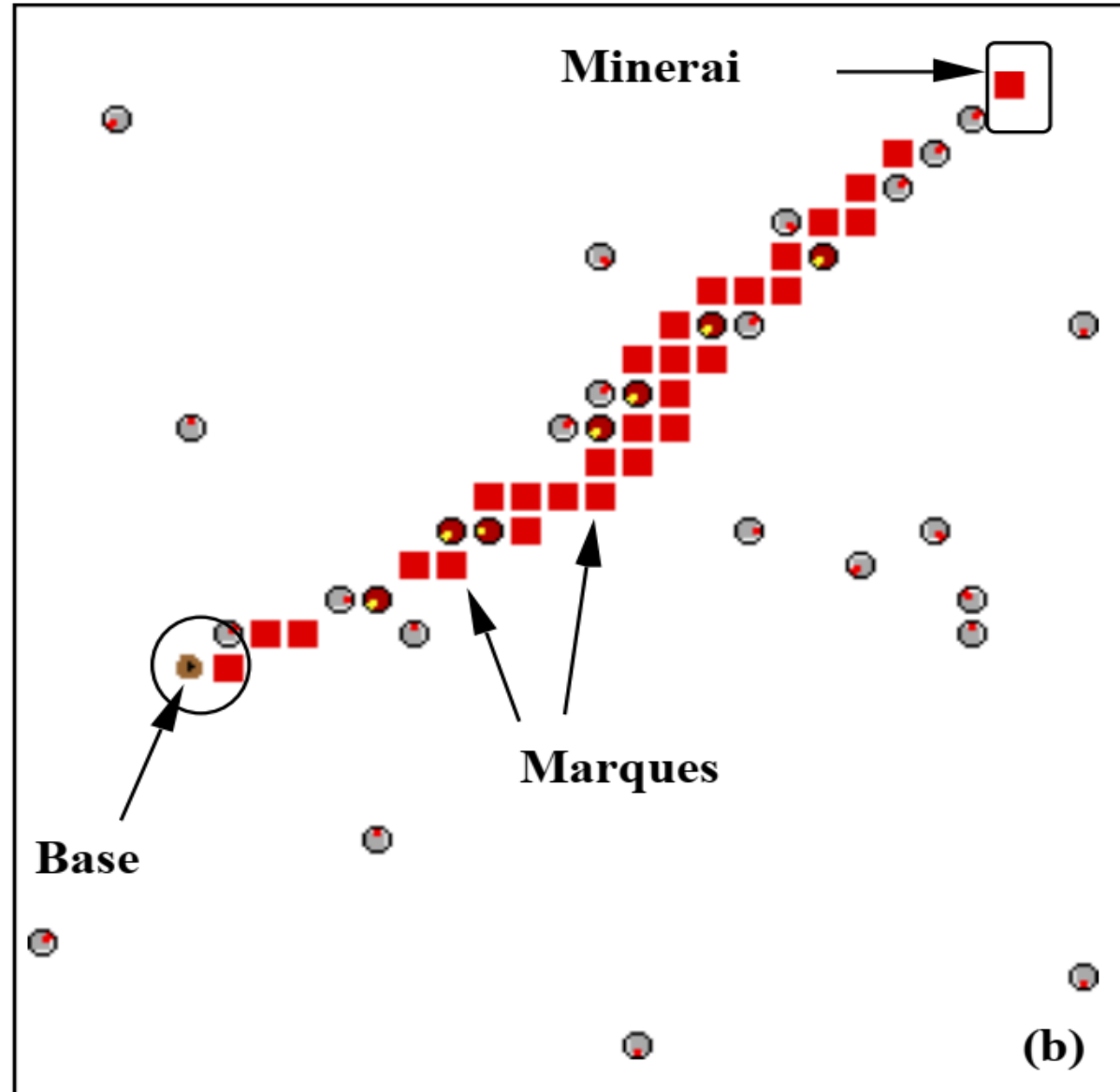
Hiérarchie des priorités de déclenchement

règle1 > règle2 > règle3bis > règle4 > règle6 > règle5

règle6 > règle 5 : le robot choisira préférentiellement de suivre une marque plutôt que de se déplacer aléatoirement.

Agents réactifs : Architecture à subsomptions (R. Brooks 1986)

Pose de marques

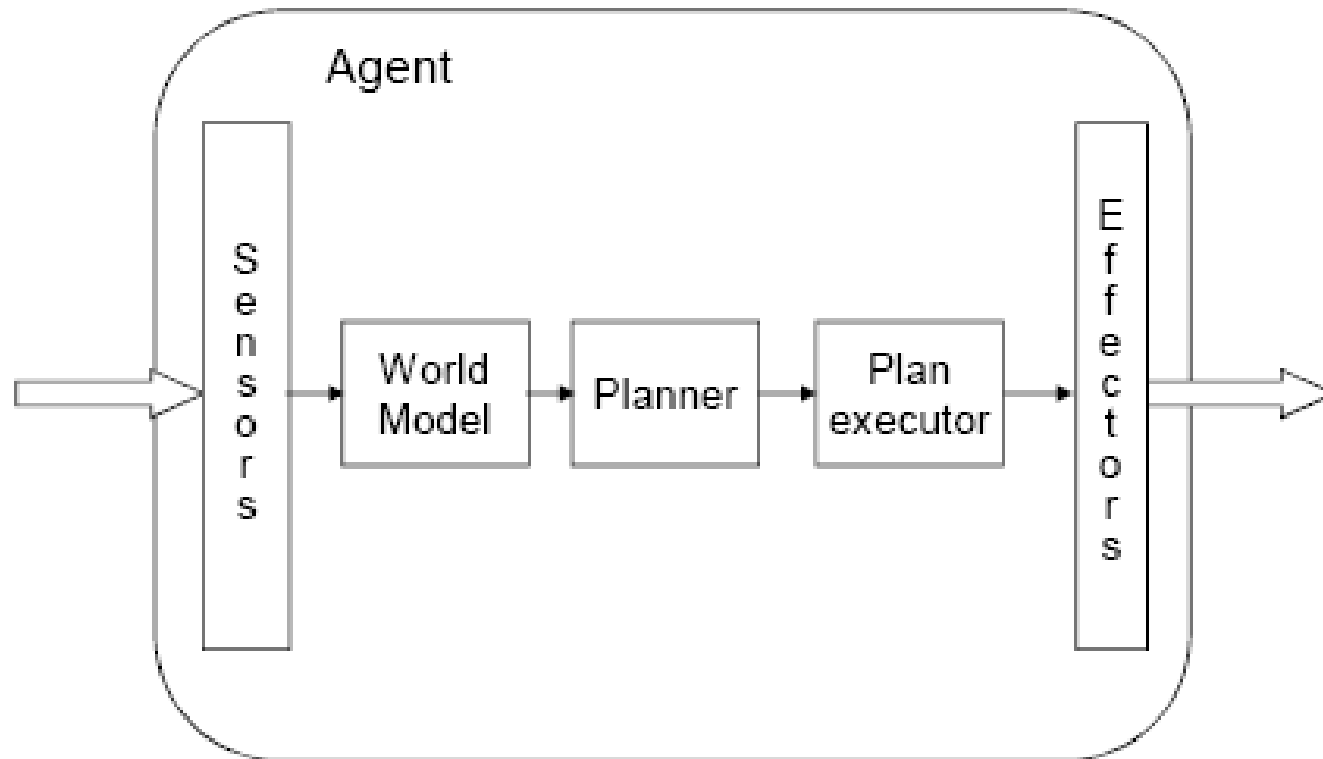


Agents cognitifs :

- ✱ 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

Agents cognitifs :

Notion de plan



Agents cognitifs :

Notion de plan

Soit E la représentation du monde actuel

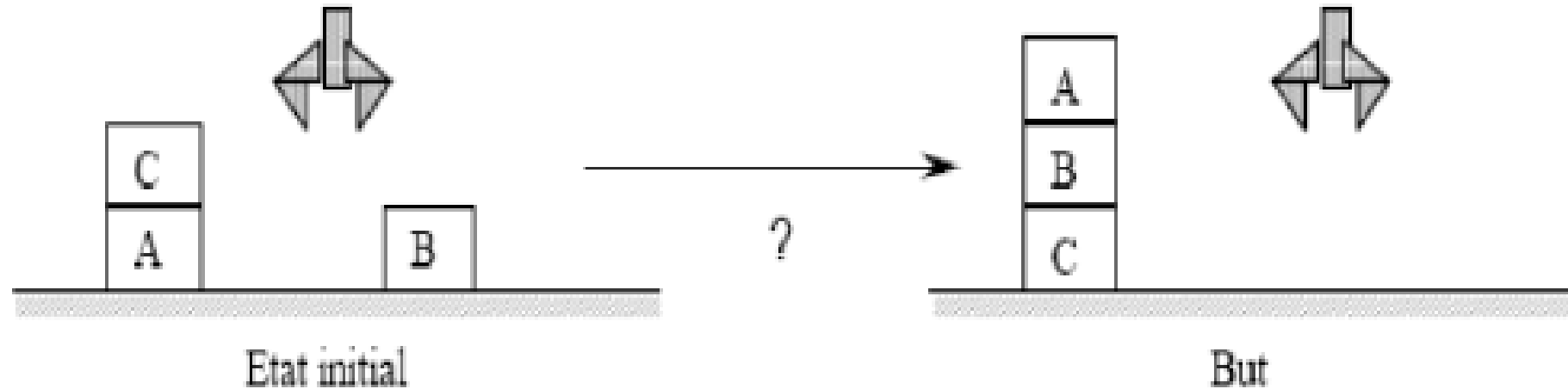
- Et soit G la représentation du monde qu'on veut atteindre
- Un problème de planification noté $(E;G)$ consiste à trouver la suite d'actions $a_1;...;a_n$

Tels que :



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

Notion de plan

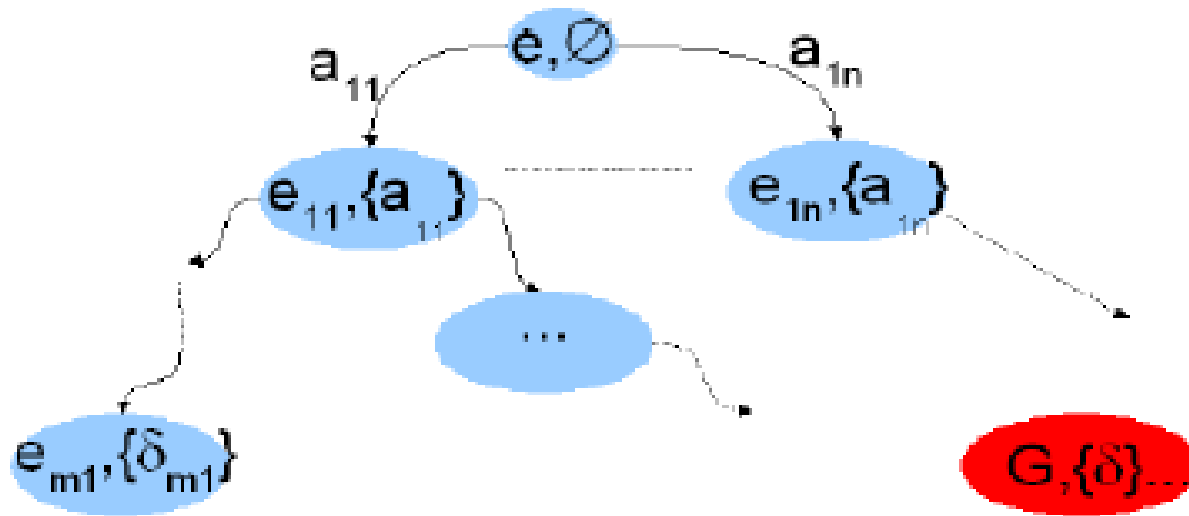


```
pickup(C) ;  
putdown(C) ;  
pickup(B) ;  
stack(B,C) ;  
pickup(A) ;  
stack(A,B)
```

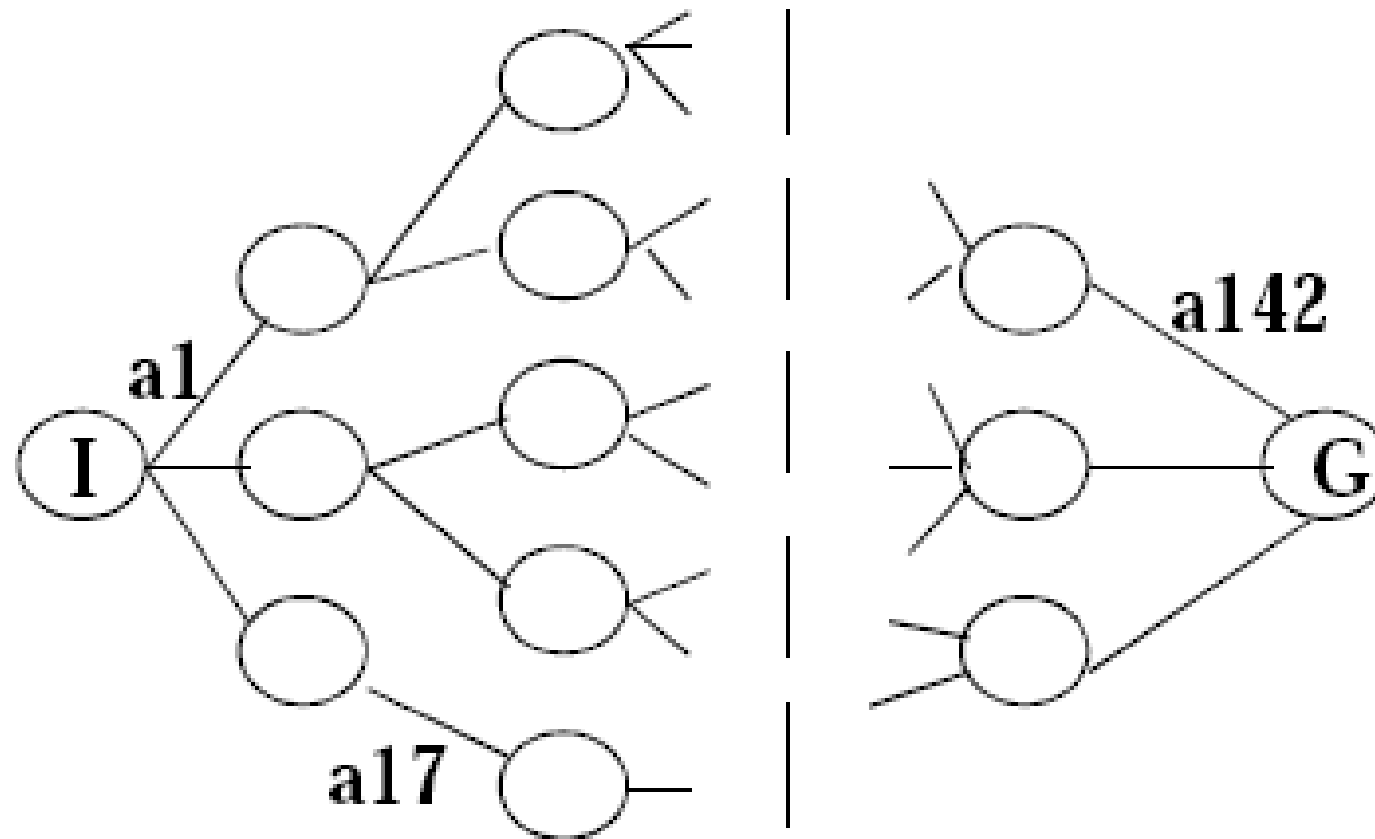
Agents cognitifs :

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.



Notion de plan



Un plan est une séquence d'actions

Agents cognitifs :

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.

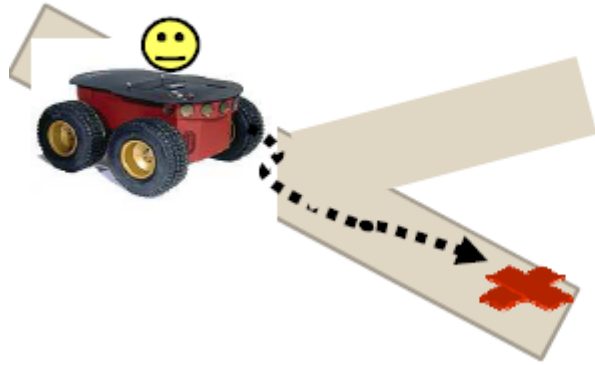
Agents cognitifs :

- Plusieurs type d'agents cognitifs:
 - Agents ayants un but
 - Agents utilisant une fonction d'utilité
 - Agent BDI (Belive, Desire, Intention)

Agents cognitifs :

- 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)

- 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.



Agents cognitifs : Architecture BDI « Beliefs, Desires, Intentions »

Une architecture BDI est conçue en partant du modèle «**Croyance-Désir-Intention**», en anglais « Belief-Desire-Intention »

Fondés sur des extensions de la logique

- Raisonnement pratique (*Practical Reasoning*) en philosophie
- issu de l'Analyse « des Fins et des Moyens » d'Aristote

Face à une décision

- Avoir certaines informations, connaissances sur le problème (« Beliefs »)
- Envisager les options possibles et les états que l'agent souhaite atteindre («Desires »), les états peuvent être contradictoires
- Choisir certains états à atteindre (« Intentions »)

Exemple d'analyse « des Fins et des Moyens » (Aristote)

BUT: Je veux emmener mon fils à l'école.

- Quelle est la différence entre ce que j'ai et ce que je veux ? une distance.
- Qu'est-ce qui change une distance ? Mon automobile. Mon automobile est en panne.
- De quoi ai-je besoin pour la faire fonctionner ? Une batterie neuve.
- Qui a des batteries neuves? Un garage. Je veux que le garage mette une batterie neuve;
- Le garage ne sais pas que je veux une batterie neuve.
- Quelle est la difficulté ? De communication.
- Qu'est-ce qui permet de communiquer ? Un téléphone ... »

❑ Analyse sous forme de séquences : de **fin**, de **fonction** nécessaire et de **moyen** qui réalise cette fonction.

Exemples:

Croyance: (Informations sur l'environnement)

Il croit qu'en travaillant dur il peut réussir

Il croit que son patient est gravement malade

Désir: (Etat final désiré)

il désire avoir son diplôme

il désire faire beaucoup de voyages

il désire avoir beaucoup d'argent

Intention: (Actions à faire pour réaliser désir)

il a l'intention de travailler (pour gagner de l'argent et pour pouvoir voyager)

il a l'intention d'étudier pour avoir son diplôme.

Le B = Belief = Croyance

Informations que l'agent possède sur l'environnement et sur d'autres agents qui existent dans le même environnement.

Les croyances peuvent être :

- incorrectes,
- incomplètes
- incertaines

Composante Révision des croyances ;

Les croyances peuvent changer en fonction de perception et/ou par l'interaction avec d'autres agents

$$\text{revc : } \mathbf{B} \times \mathbf{P} \text{ ----} \rightarrow \mathbf{B}$$

P: représente l'ensemble des perceptions de l'agent .

Le D = Desire = Désir

Les désirs d'un agent représentent les états de l'environnement, et parfois de lui-même, que l'agent aimerait voir réalisés.

Un agent peut avoir des désirs contradictoires; il doit alors, choisir parmi ses désirs un sous-ensemble consistant qui est parfois identifié avec les buts de l'agent.

Composante Processus de décision

C'est la fonction qui peut changer les désirs d'un agent si ses croyances ou intentions changent.

options : B x D x I ----> D

Le I = Intention = **Intention**

Les intentions d'un agent sont les désirs que l'agent a décidé d'accomplir ou les actions qu'il a décidé de faire pour accomplir ses désirs.

□ l'agent peut ne pas être capable d'accomplir tous ses désirs à la fois.

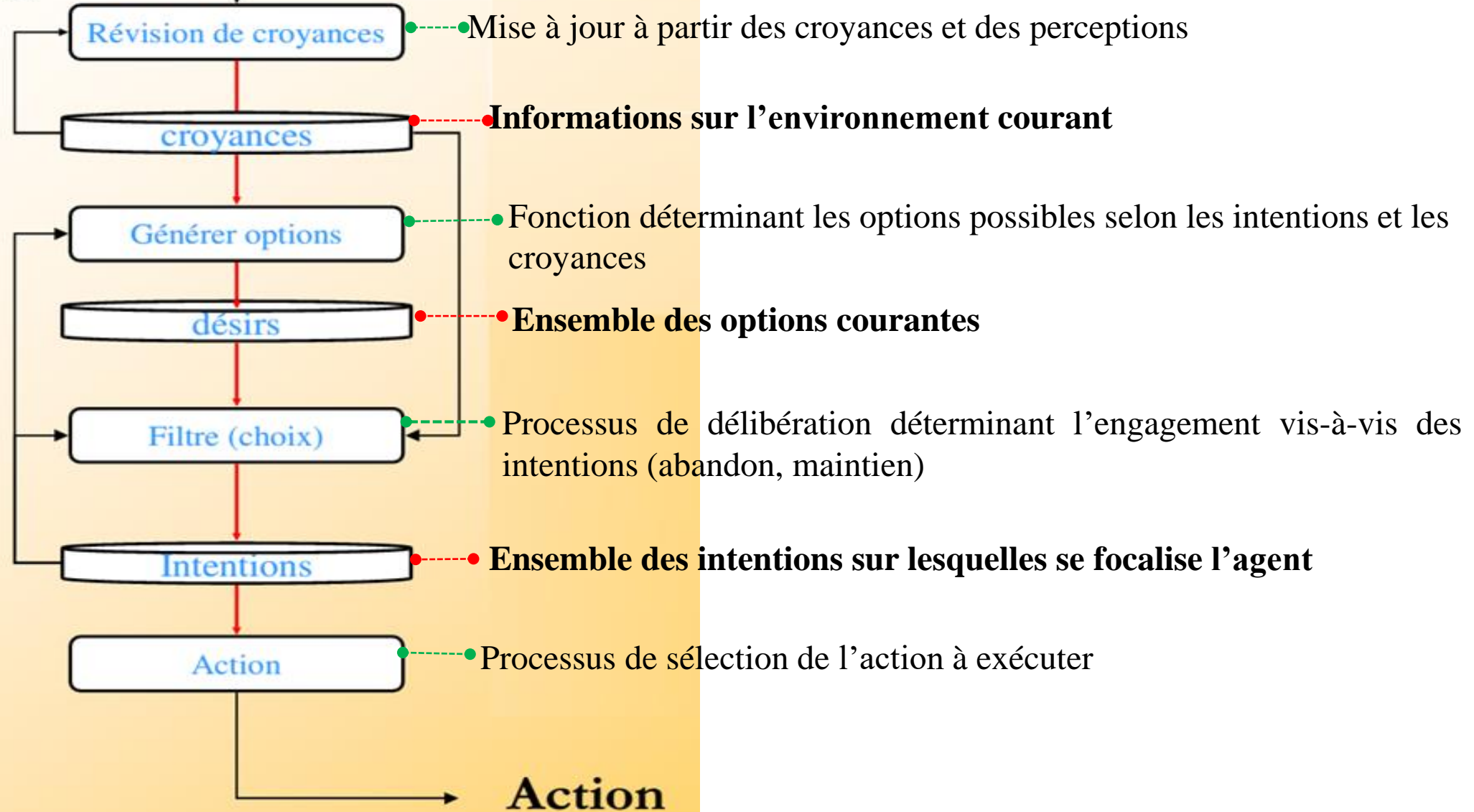
Processus de décision (composante Filtre)

est la fonction qui représente le processus de décision de l'agent prenant en compte ses désirs et ses intentions courantes.

$$\textit{filtre} : B \times D \times I \rightarrow I$$

Filtre est la fonction la plus importante car elle décide des intentions à poursuivre.

**Perceptions,
communications**



Architecture BDI d'un agent

Algorithme de contrôle d'agent BDI

$B = B_0$

$D = D_0$

$I = I_0$

Début

$B := revc(B ; p)$

$D := options(D ; I)$

$I := filtre(B ; D ; I)$

$A := plan(B, I)$

$execute(A)$

Fin.