

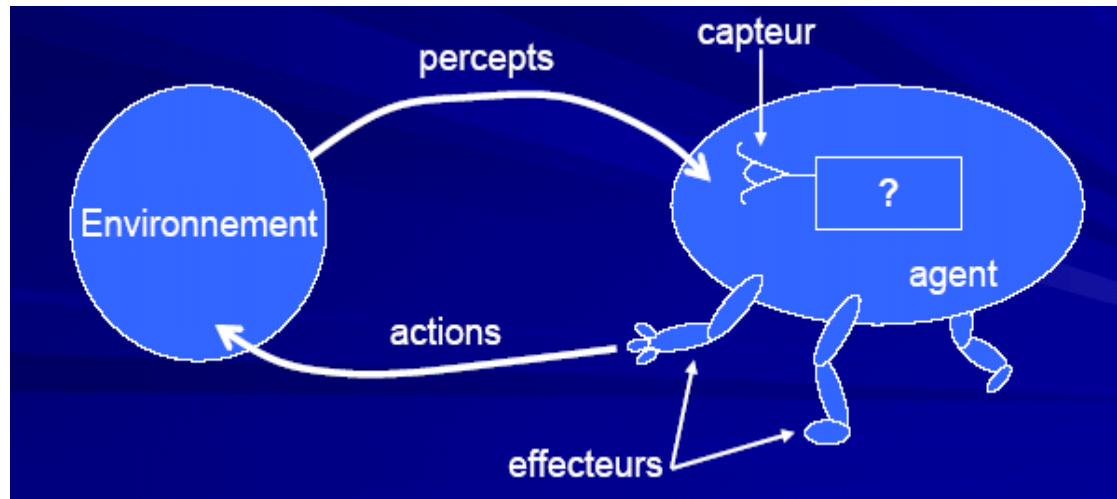
Agents intelligents

Exercices-Démos

Une définition

Agent Intelligent:

tout ce qui **Perçoit** son **Environnement** à l'aide de ses **Capteurs** et qui **Agit** sur son environnement l'aide de ses **Effecteurs**..



Caractéristiques d'un agent

- **Situé**
- **Autonome**
- **Proactif**
- **Adaptatif**
- **Mobile**
- **Rational**
- **Intelligent**
- **Social**

l'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs: Coopération, concurrence,

-

Structure d'un agent

4 types de base en ordre de généralité grandissante:

–Agent simple réflexe

–Agent réflexe avec état interne

–Agent basé sur les buts

–Agent basé sur l'utilité

Réactif ↑

Cognitif ↓

La mesure de performance

Actions \Rightarrow $R+$
 a \Rightarrow v

Il faut faire attention à ce qu'on choisit.

Une meilleure option (exemple: Aspirateur):

- Récompenser l'agent pour un plancher propre
- Peut-être avec une pénalité pour l'électricité consommé).

Agent basé sur une utilité

- préférer les états qui ont une grande valeur pour la fonction d'utilité
- fonction d'utilité

$$u : S \rightarrow D$$

avec

D = domaine des mesures d'utilité

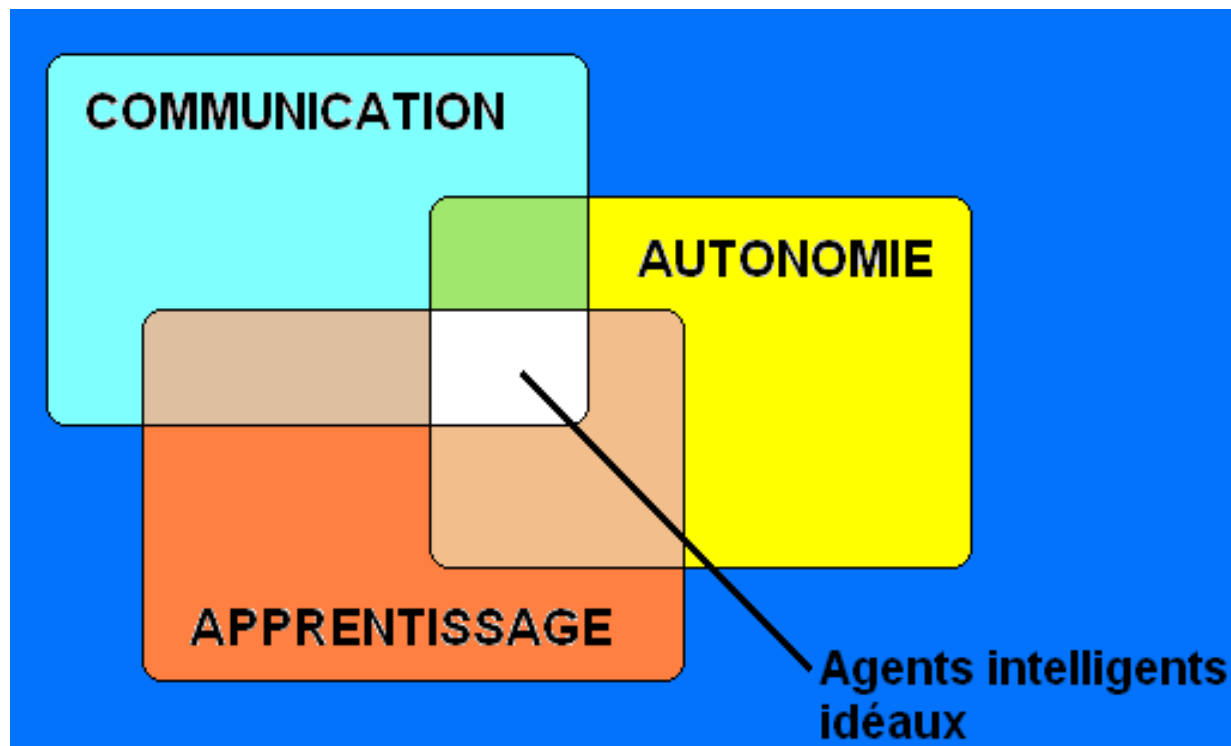
S = ensemble d'états

- décisions possibles:
 - choix entre des objectifs en conflit
 - choix entre la vraisemblance de succès et l'importance de l'objectif

Propriétés de l'environnement

- Complètement observable vs partiellement observable:
- Déterministe vs stochastique:
- Épisodique vs séquentielle:
- Statique vs dynamique:
- Discret vs continu:
- Un agent vs multi-agent:

Caractéristiques des agents intelligents



Agent rationnel

Rationnel = basé sur le raisonnement

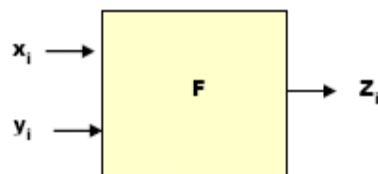
Agent rationnel: l'agent doit exécuter l'action qui maximise sa mesure de performance en fonction de sa perception du monde et de ses connaissances.

« 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

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 !

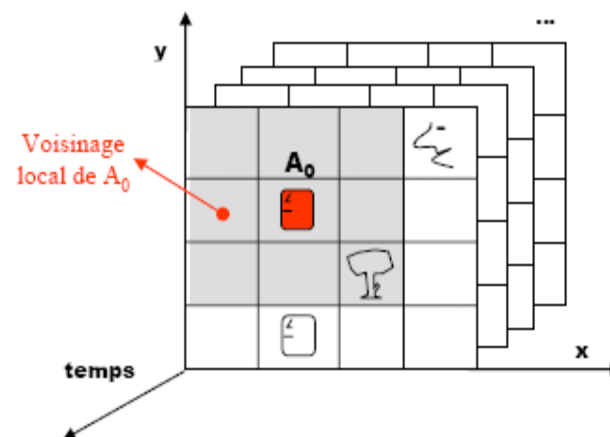
```
F[x_] := (CPT++; x+CPT)
```

```
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.

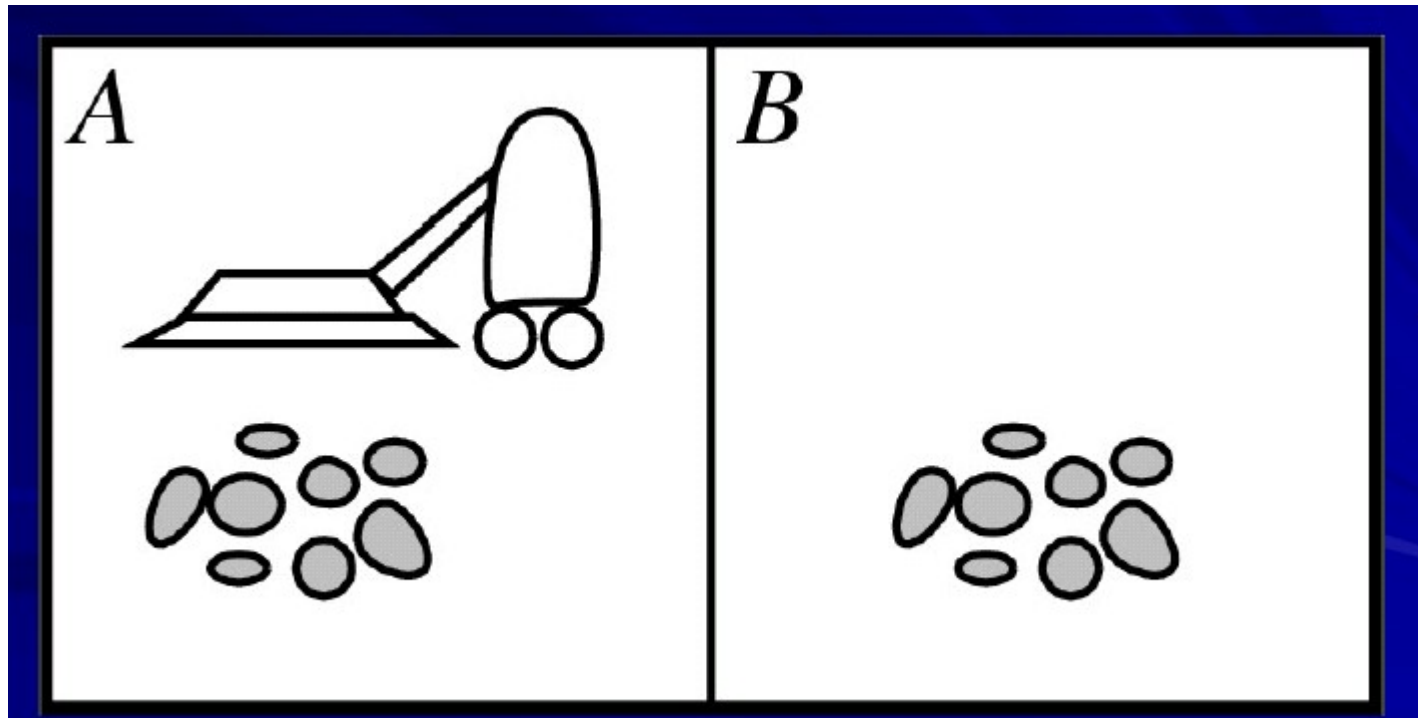
Un agent peut être constitué de plusieurs objets.

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

Objets	Agents
Classe	Rôles, Types
Message	Assertion, Requête, Commande, Changement d'état, Evénement
Attribut	Etat, Croyances, Objectifs, Meta connaissance
Méthode	Plans, Capacités, Règles, 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 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);
```

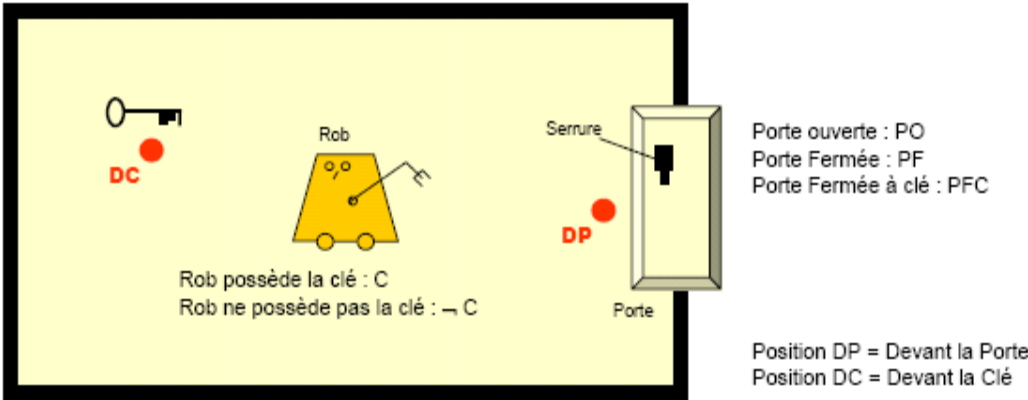
```
regle=regle-correspondre(etat,regles)
```

```
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 \wedge PO	Alors SORTIR
Si DP \wedge PF	Alors OUVRIR
Si DP \wedge PFC \wedge C	Alors DEVEROUILLER
Si DP \wedge PFC	Alors RANDOM-WALK
Si DC \wedge C	Alors RANDOM-WALK
Si DC \wedge ¬ 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

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

Base de règles (BR) :

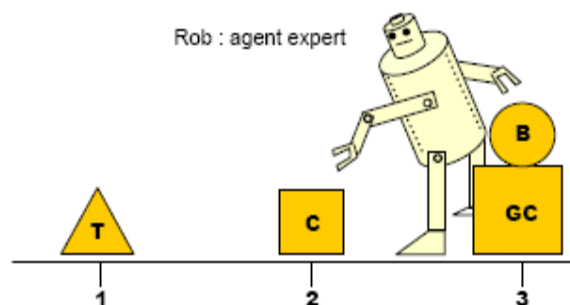
Les règles sont de la forme : **If** <condition-list> **Then** <action-list>

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

If $\text{free}(T) \wedge \text{free}(GC)$ **then** $\text{PUTON}(T, GC)$
If $\neg \text{free}(T)$ **Then** $\text{MOVETO}(X | \text{upon}(T, X), Y | \text{free}(Y))$
If $\neg \text{free}(GC)$ **Then** $\text{MOVETO}(X | \text{upon}(GC, X), Y | \text{free}(Y))$

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



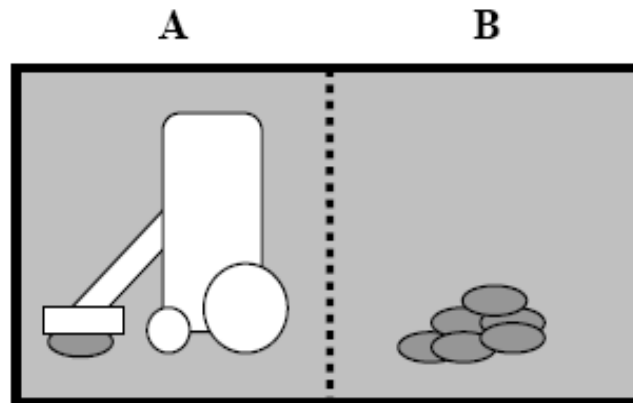
Le monde de cubes

Terry Winograd (1972)

EXEMPLES

EXEMPLE 1

On étudie un agent intelligent prenant la forme d'un aspirateur robotisé. Son environnement se compose de deux lieux communicants, *A* et *B*. Chaque lieu peut être initialement dans 2 états possibles : propre ou sale. L'agent perçoit sa position (dans quel lieu il se trouve), et l'état de ce lieu. Il peut effectuer les actions suivantes : aller à *A*, aller à *B*, aspirer la saleté, relâcher la saleté, ne rien faire. Initialement, la position de l'agent et les états des lieux sont aléatoires. Le but de l'agent est que le monde entier (c'est-à-dire *A* et *B*) soit propre.



Exercice 1 (solution)

Quelles sont les propriétés de l'environnement (observable ? déterministe ? etc.) ?

Exercice 2 (solution)

On définit une mesure de performance correspondant au nombre de fois que l'agent a aspiré de la saleté. Cela signifie que l'on considère que plus l'agent aspire de saleté, plus il se rapproche de son but.

Donnez un algorithme simple pour que l'agent soit rationnel selon cette mesure. S'agit-il d'une bonne mesure de performance, et pourquoi ?

Exercice 3 (solution)

Proposez une meilleure mesure, et donnez un algorithme simple pour que l'agent soit rationnel selon cette mesure.

Exercice 4 (solution 4 et 5)

On modifie l'environnement : les lieux peuvent se salir, c'est-à-dire que de la saleté peut apparaître aléatoirement. Que deviennent les propriétés de l'environnement ?

Exercice 5

Donnez un algorithme tel que l'agent évoluant dans le nouvel environnement et utilisant la mesure de performance que vous avez proposée soit rationnel.

Solution Exercice 1

Propriétés de l'environnement :

- **partiellement observable** : puisque l'agent ne connaît pas l'état du lieu où il n'est pas
- **déterministe** : le prochain état du monde ne dépend que des actions de l'agent
- **séquentiel** : le passé influence le futur (si l'aspirateur nettoie, le lieu devient propre)
- **statique** : seules les actions de l'agent modifient le monde
- **discret** : tout est géré de façon discrète (lieu : *A* ou *B*, pas de position intermédiaire ; état du lieu : propre ou sale, pas d'intermédiaire ; etc.)
- **mono-agent**

Solution Exercice 2

si le lieu actuel est propre

si le lieu actuel est \mathcal{A} alors

aller à \mathcal{B}

sinon

aller à \mathcal{A}

fin si

fin si

répéter

nettoyer

relâcher

fin répéter

Un comportement maximisant cette mesure consisterait à rester dans le même lieu et aspirer/relâcher la même saleté indéfiniment. Un agent rationnel par rapport à cette mesure ne sera donc pas très efficace pour qui, l'objectif de nettoyer \mathcal{A} et \mathcal{B} .

Solution Exercice 3

Soit la mesure correspondant au nombre de lieux propres dans le monde :
cette mesure oblige l'agent à nettoyer et à se déplacer, et semble donc
meilleure que la précédente.

répéter deux fois
 si lieu actuel est sale alors nettoyer
 si le lieu actuel est \mathcal{A} alors
 aller à \mathcal{B}
 sinon
 aller à \mathcal{A}
 fin si
fin répéter

Solution Exercice 4

Propriétés de l'environnement qui sont modifiées :

- **stochastique** : le monde est modifié aléatoirement
- **dynamique** : le monde peut être modifié même si l'agent ne fait rien

Solution Exercice 5

On reprend l'algorithme précédent en répétant indéfiniment la boucle :

répéter

si le lieu actuel est sale alors nettoyer

si le lieu actuel est \mathcal{A} alors

aller à \mathcal{B}

sinon

aller à \mathcal{A}

fin si

fin répéter

EXEMPLE 2

Exercice:

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.

EXEMPLE 2

Solution:

Percepts : Pixel d'intensité variable (caméra), cote du livre (lecteur de code barre).

Actions :

- Prendre un livre, s'il y a un nouveau livre à ranger.
- Lire la cote.
- Se déplacer (avancer, tourner, arrêter) pour aller ranger un livre ou revenir au chariot des retours et pour éviter les obstacles.
- Déposer un livre lorsqu'il est arrivée à la position où le livre doit être rangé.

But : Le but de l'agent est de mettre les livres, qui sont dans le chariot des retours, à la bonne position dans les rayons de la bibliothèque.

Environnement : L'environnement de l'agent est une bibliothèque.

L'environnement est *inaccessible*, parce que l'agent ne peut pas avoir accès à toutes les informations sur l'environnement.

L'environnement est *non-déterminé*, parce que les actions de l'agent n'ont pas un effet garanti et il y a de l'incertain. Lorsque l'agent avance, il peut heurter un obstacle qu'il n'avait pas vu.

L'environnement est *non-épisodique*, car les actions de l'agent dans le passé peuvent influencer ses actions futures. Par exemple, s'il doit replacer un deuxième livre, il va pouvoir tenir compte des obstacles qu'il avait rencontrés lors du rangement du livre précédent. Cela va influencer le chemin qu'il va emprunter.

L'environnement est *dynamique*, parce qu'il y a une multitude d'événements qui peuvent se produire pendant que l'agent délibère. Un nouvel obstacle à éviter, un nouveau livre à ranger, etc.

L'environnement est *continu*, car il y a plusieurs perceptions possibles et plusieurs actions possibles.

EXEMPLE 3

Exercice

Pour chacun des exemples d'agents suivants, dites quelle architecture d'agent est la plus appropriée (simple réflexe, réflexe avec état interne, but et utilité) et pourquoi ?

- a. Un agent contrôlant une valve de pression d'une centrale nucléaire.
- b. Un agent devant sortir d'un labyrinthe.
- c. Un agent conduisant une automobile.
- d. Un agent qui achète et vend des actions sur Internet.
- e. Un agent qui nettoie la vaisselle et qui la range dans les armoires.

EXEMPLE 3

Solution

- a. Simple-réflexe : l'agent ne fait qu'obéir à des règles simples. Par exemple, si la pression dépasse une certaine valeur, ouvrir la valve.
- b. But : L'agent a le but de sortir du labyrinthe.
- c. Utilité : Il y a plusieurs alternatives et de l'incertain.
- d. Utilité : Il y a des décisions à prendre entre la sécurité et le rendement et il y a de l'incertain.
- e. Réflexe : Un agent réflexe est suffisant pour cette tâche, (mais aussi réflexe avec état interne est accepté si on tient compte que l'agent doit se souvenir si les armoires sont pleines.

EXEMPLE 4

Exercice

Donnez la PEAS (Performance, Environment, Actuators, Sensors : Performance Environnement, Actions, Percepts) et les propriétés de l'environnement pour un agent responsable des lumières à une intersection. Quelle architecture d'agent utiliseriez-vous pour ce type d'agents et pourquoi?

Solution

- **Mesure de performance** : Le temps d'attente moyen des autos et des piétons.
- **Environnement** : Les autos, les lumières, les routes, les piétons.
- **Effecteur** : Les lumières pour les autos et pour les piétons.
- **Capteur** : Pesé sur la route et bouton pour les piétons.

Propriétés de l'environnement : Partiellement observable, stochastique, séquentiel, dynamique, continu, mono agents.

Architecture : Agent basé sur l'utilité. La fonction d'utilité doit tenir compte du temps moyen d'attente des autos et des piétons.

Questions-Réponses

1 Est-ce qu'un système doit penser comme un humain pour passer le test de Turing? Expliquez.

Non, il n'a seulement qu'à agir comme un humain.

2 Est-ce qu'un agent qui a une perception partielle de l'environnement peut être parfaitement rationnel ? Expliquez.

Oui, la rationalité et l'omniscience sont des concepts différents. Un agent n'a pas besoin de tout percevoir pour pouvoir agir rationnellement. Il doit percevoir tout ce qu'il peut et agir par la suite au meilleur de ses connaissances.

3 Est-ce que l'affirmation suivante est vraie : un agent rationnel est meilleur que tous les agents non rationnels parce qu'il sait le résultat réel de ses actions ? Expliquez.

Non, dans un environnement inaccessible et/ou stochastique, un agent rationnel ne peut pas savoir les résultats réels de ses actions. Dans ce type d'environnement, s'il n'est pas chanceux, il pourrait être moins bon qu'un agent non rationnel.

4 Donnez la PEAS et les propriétés de l'environnement pour un robot d'inspection sur Mars. Le robot doit trouver des échantillons intéressants, les analyser et transmettre ses analyses sur Terre.

Mesure de performance : Nombre d'échantillons différents analysés, la qualité des échantillons, minimiser le temps de déplacement, la surface parcourue, etc.

Environnement : Sable, roches, autres robots, vent, poussière, etc. En fait, tout l'environnement de la planète Mars.

Effecteurs : Moteurs, pince, outils pour l'inspection, émetteur radio, etc.

Capteurs : Caméras, récepteur radio, outils d'analyse d'échantillon, etc.

Propriétés de l'environnement : Partiellement observable, stochastique, séquentiel, dynamique, continu et multiagent (si on considère qu'il y a plusieurs robots)