

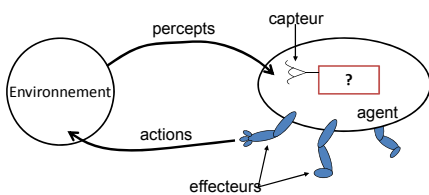
Agents Intelligents

Plan

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

Qu'est-ce qu'un agent intelligent?

- **Agent intelligent**: un entité qui perçoit son **environnement** à l'aide de ses **capteurs** et qui agit sur son environnement à l'aide de ses **effecteurs**.

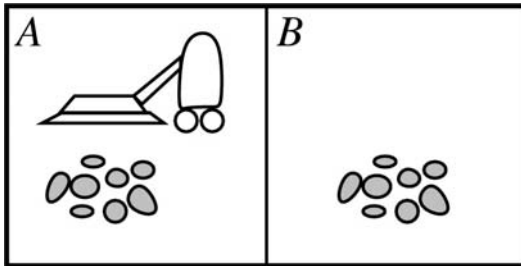


Agents vs Objets

- **Agent** : entité **autonome** interagissant avec son environnement
- **Objet** : entité passive possédant un état et sur lequel on peut effectuer des opérations.
- Un agent est à un degré d'abstraction plus élevé qu'un objet.
 - 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)

Exemple très simple d'agent

- Un agent aspirateur



Agent rationnel

- 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**.
- Mesure de performance
 - Externe
 - Fixée par le concepteur
 - Propre à la tâche

Choix de la mesure de performance

- Il faut faire attention à ce qu'on choisit
 - Ex: Quantité de saleté ramassé en 8 heures
 - L'agent pourrait ramasser la saleté, la redéposer et ainsi de suite.
- 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 consommée).

Agent rationnel

- Rationnel ≠
 - Omniscient
 - Clairvoyant
 - Succès
- Rationnel ⇒
 - 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 deux : agents et non-agents.

Environnement de la tâche

- La première étape lors de la conception d'un agent est de spécifier l'environnement de la tâche (task environment) qui contient les quatre éléments suivants (PEAS) :
 - Mesure de la performance (Performance measure)
 - Environnement (Environment)
 - Effecteurs (Actuators)
 - Capteurs (Sensors)

Exemple

- Conducteur de taxi automatique
 - **Mesure de performance** : sécurité, vitesse, légalité, confort, profits, etc.
 - **Environnement** : routes, autres autos, piétons, clients, etc.
 - **Effecteurs** : volant, accélérateur, frein, clignotant, klaxon, etc.
 - **Capteurs** : caméras, GPS, sonar, odomètre, indicateur de vitesse, capteurs du moteur, etc.

Exemple du Taxi (suite)

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figure 2.4 PEAS description of the task environment for an automated taxi.

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 ?

Propriétés de l'environnement

- **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 l'environnement change avec le temps ?

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 multiagent** : Est-ce qu'il y a un ou plus qu'un agent, et que ces agents interagissent ensemble ?

Propriétés de l'environnement

- **Situation la plus difficile** :
 - Partiellement observable
 - Stochastique
 - Séquentielle
 - Dynamique
 - Continue
 - Multiagent
- **Exemple, conduite automatisée d'un taxi**

Exemple						
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

Exemple (2)

- **Déterministe/Stochastique ou Stratégique :**
Dans ce cas-ci l'état est conjoint est il est déterminé par l'état conjoint actuel et l'action conjointe des agents
- **Statique/dynamique ou Semi :**
dans ce cas ci l'environnement ne change pas avec le temps, mais la performance de l'agent, elle si.

Structure d'un agent

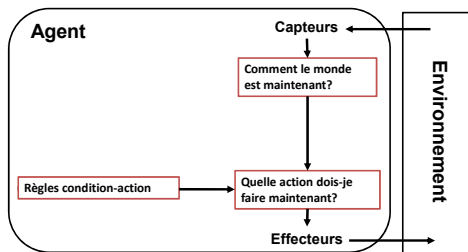
- 4 types de base en ordre de généralité grandissante :
 - Agent simple réflexe
 - Agent réflexe avec état interne
 - Agent basé sur les buts
 - Agent basé sur l'utilité
- Tous ces types peuvent être transformés en agents apprenants.

Agent simple réflexe

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

```
function REFLEX-VACUUM-AGENT([location, status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

Agent simple réflexe

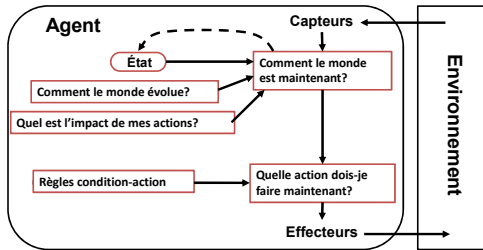


Agent simple réflexe

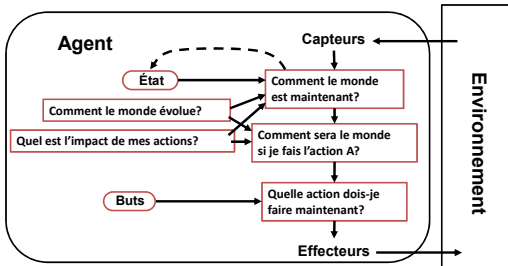
```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  static: rules, a set of condition-action rules
  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION(rule)
  return action
```

Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Agent réflexe avec état interne



Agent basé sur les buts

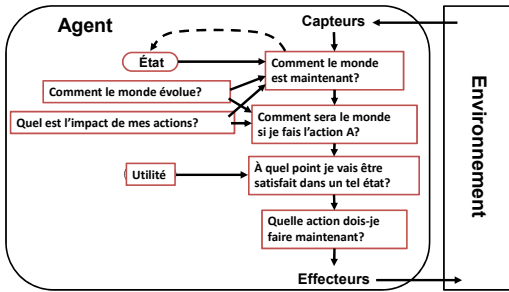


Facile si une action mène directement au but; sinon, il faut utiliser un algorithme de recherche ou de planification. Plus flexible qu'un agent réflexe, car on peut changer de but sans réécrire toutes les règles.

Agent basé sur l'utilité

- Les buts ne font la distinction que entre un état « heureux » ou « pas heureux ».
- Fonction d'utilité: état \rightarrow nombre (valeur)
- Aide dans deux cas où les buts échouent :
 - Buts en conflits (ex: vitesse et sécurité)
 - Lorsqu'il y a plusieurs buts

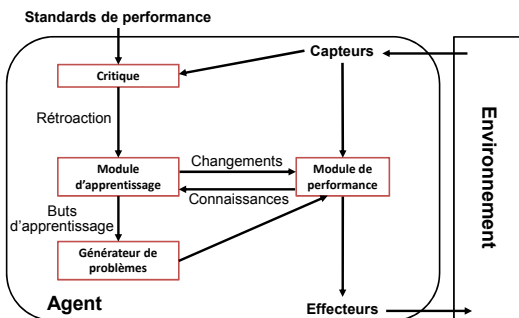
Agent basé sur l'utilité



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

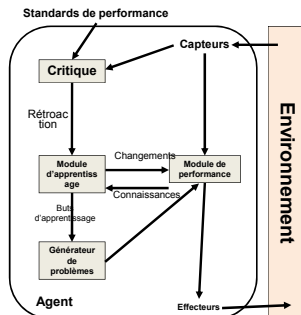


Agent taxi apprenant

- **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.

Agent taxi 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.



Agent taxi apprenant

- ▶ **Module de performance**
 - ▶ le taxi fait des actions sur la route : **Conduit n'importe comment**
- ▶ **Critique**
 - ▶ **Observe le mécontentement des autres conducteurs; il avertit alors le module d'apprentissage.**
- ▶ **Module d'apprentissage**
 - ▶ **Élabore une règle disant que c'est une mauvaise action et le module de performance est modifié pour une autre règle.**
- ▶ **Générateur de problèmes**
 - ▶ **Détecte un besoin d'amélioration est suggère d'expérimenter d'autres façons de faire (conduite sur la neige)**
