

IFT 615 – Intelligence artificielle

Agents intelligents

Hugo Larochelle

Département d'informatique

Université de Sherbrooke

<http://www.dmi.usherb.ca/~larocheh/cours/ift615.html>

Sujets couverts

- Intelligence artificielle
- Agents intelligents
- Rationalité
- PEAS (mesure de Performance, Environnement, Actionneurs, Senseurs)
- Types d'environnements
- Types d'agents
- Exemple – Le monde des wumpus (Wumpus world)

Intelligence

- Avant de définir l'intelligence artificielle, il faut définir l'intelligence !
- Définition ?

Intelligence Artificielle (IA)

- **Définition :**

« Branche de l'informatique ayant pour objet l'étude du traitement des connaissances et du raisonnement humain, dans le but de les reproduire artificiellement pour ainsi permettre à un appareil [(agent)] d'exécuter des fonctions normalement associées à l'intelligence humaine. »

[Grand dictionnaire terminologique, 2006]

Deux branches de l'IA

- **Compréhension de l'intelligence**

- ◆ Neurosciences computationnelles

- » Développer des modèles mathématiques du fonctionnement du cerveau au niveau neuronal

- ◆ Sciences cognitives, psychologie

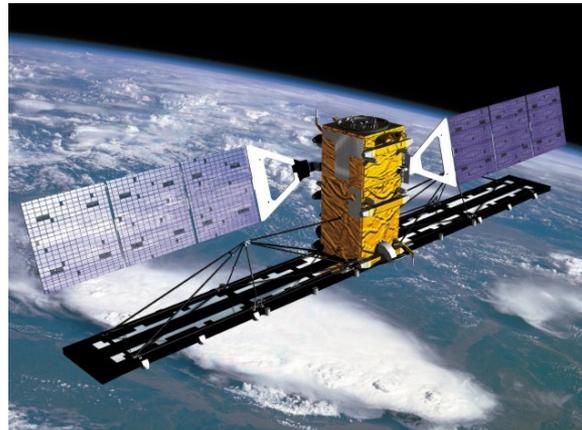
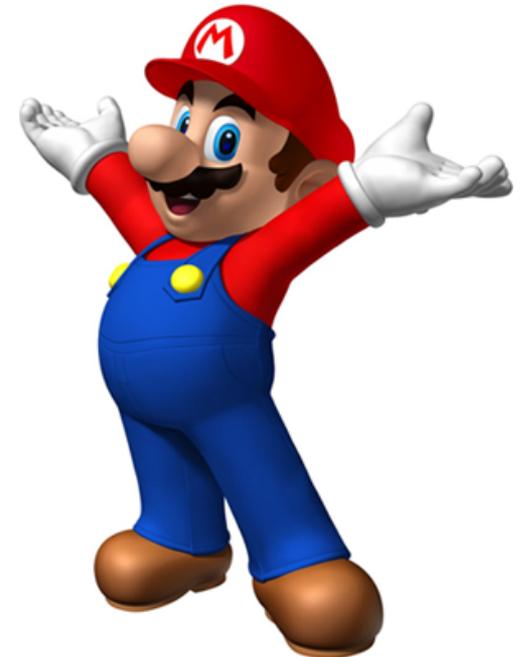
- » Comprendre le raisonnement humain
 - » Prédire la performance d'un humain à une tâche
 - Ex: l'architecture ACT-R pour évaluer le risque couru en parlant au téléphone lors de la conduite d'une voiture (modèle de *multitasking* chez l'humain)

- **Création d'agents intelligents**

- ◆ Capacités fondamentales:

- » Perception
 - » Représentation des connaissances (modélisation)
 - » Apprentissage
 - » Raisonnement
 - » Prise de décisions

Exemples d'agents intelligents

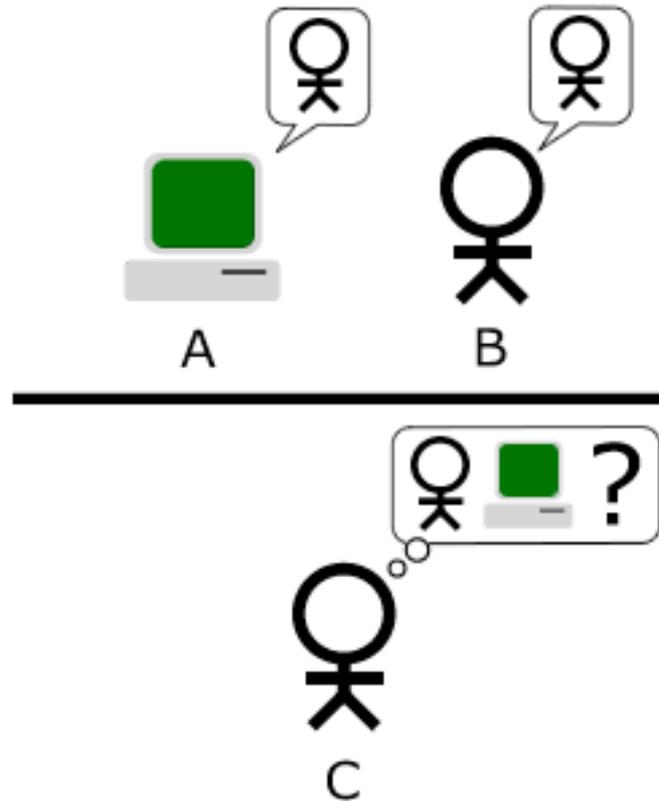


(1) Système d'aide à la décision; (2) Azimut-3; (3) Rover de la NASA;
(4) Radarsat-II de l'ASC; (5) Mario de Nintendo.

Pourquoi avoir une IA ?

- Programmation d'actions vs Décisions automatiques
- **Programmation d'actions**
 - ◆ Scripts
 - ◆ Machine à états finis
- **Décisions automatiques** (~ programmation dynamique)
 - ◆ Les actions à exécuter ne sont ni scriptées, ni programmées à l'avances
 - ◆ L'agent décide lui-même de ses propres actions, à partir d'un certain calcul ou « raisonnement »
 - ◆ On donne à l'ordinateur la **capacité** de prendre des décisions intelligentes, dans toute situation possible

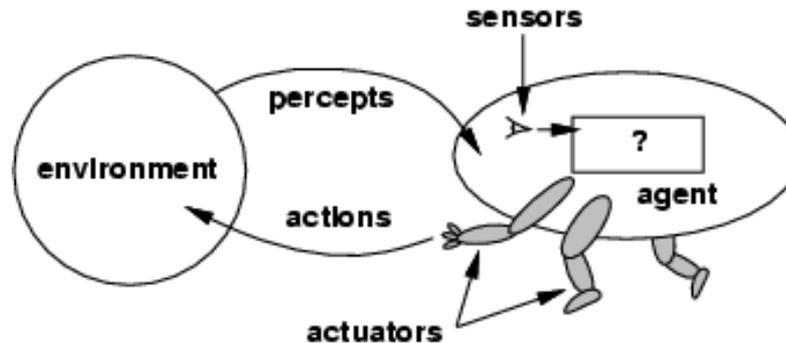
Test de Turing



Agents

- Un agent est n'importe quel entité qui perçoit son environnement par des **senseurs** (capteurs) et agit sur cet environnement par des **actionneurs** (*actuators*)
- Un agent humain a:
 - ◆ des yeux, des oreilles, et d'autres senseurs
 - ◆ des mains, des jambes, une bouche et d'autres actionneurs
- Un agent robot a:
 - ◆ des caméras, des senseurs infra rouges et autres senseurs
 - ◆ des roues, des jambes, des bras-articulés, et d'autres actionneurs
- Un agent logiciel a:
 - ◆ un clavier, un accès lecture à un disque dur et autres senseurs
 - ◆ un écran, un accès écriture à un disque dur comme actionneurs

Agents et environnements



- Le **processus agent** f prend en entrée une séquence d'**observations** (percepts) et retourne une **action**:

$$[f: P^* \rightarrow A]$$

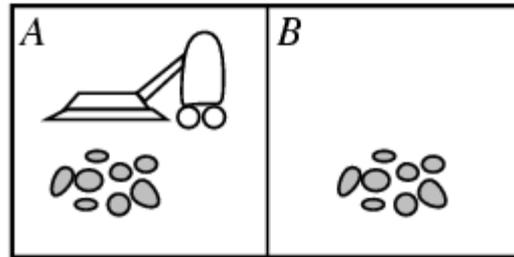
- En pratique le processus est un implémenté par un programme sur une architecture matérielle particulière

Ébauche d'un agent

```
function SKELETON-AGENT(percept) returns action
  static: memory, the agent's memory of the world

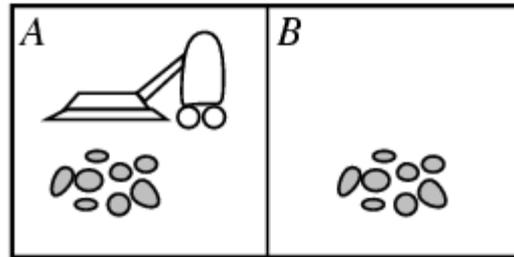
  memory ← UPDATE-MEMORY(memory, percept)
  action ← CHOOSE-BEST-ACTION(memory)
  memory ← UPDATE-MEMORY(memory, action)
  return action
```

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,Clean] → Right

[A,Dirty] → Suck

...

[A,Clean] [A,Clean] [A,Dirty] → Suck

[A,Clean] [A,Clean] [A,Clean] → Right

...

Agents rationnels

- Un **agent rationnel** doit agir “correctement” en fonction de ce qu’il perçoit et de ses capacités d’action:
 - ◆ l’action correcte est celle permettant à l’agent de réussir le mieux
- **Mesure de performance:**
 - ◆ une fonction objective mesurant la qualité d’un comportement de l’agent
- Par exemple, une mesure de performance pour le robot aspirateur pourrait être:
 - ◆ la quantité de déchets aspirés
 - ◆ la propreté des lieux
 - ◆ la durée de la tâche
 - ◆ le bruit généré
- Agent rationnel: Étant donné une séquence d’observations (données sensorielles) et des connaissances propres, un agent rationnel devrait choisir une action qui maximise la mesure de performance

Agents rationnels

- Rationalité ne veut pas dire « qui sait tout »
(par exemple, connaît tous les effets de ses actions)!
- Rationnel ne veut pas dire « parfait »
 - ◆ La rationalité maximise la performance espérée
 - ◆ La perfection maximise la performance réelle/actuelle
 - ◆ Mais souvent on ne peut pas connaître la performance réelle avant l'action
- Un agent peut effectuer des actions d'observation pour cueillir des informations nécessaires à sa tâche
- Un agent est **autonome** s'il est capable d'adapter son comportement en fonction de son expérience (capacité d'apprentissage et d'adaptation)

Modèle PEAS

- PEAS: Un modèle de conceptions des agents par la spécification des composantes majeures suivantes:
 - ◆ Mesure de performance (Performance)
 - ◆ Connaissance de l'environnement (Environnement)
 - ◆ Les actions que l'agent peut effectuer (Actionneurs)
 - ◆ La séquence des observations ou percepts de l'agent (Senseurs)
- PEAS = Performance, Environnement, Actionneurs, Senseurs

Modèle PEAS pour un robot taxi

- **Agent:** robot taxi
- **Mesure de performance:** sécurité, vitesse, respect du code routier, voyage confortable, maximisation des profits
- **Environnement:** route, trafic, piétons, clients
- **Actionneurs:** volant, changement de vitesse, accélérateur, frein, clignotants, klaxon
- **Senseurs:** caméras, sonar, speedometer, GPS, odomètre, témoins du moteur, etc.

Modèle PEAS pour un diagnostique médical automatisé

- **Agent:** système de diagnostique médical
- **Mesure de performance:** santé des patients, minimisation des coûts, satisfaction des patients
- **Environnement:** patients, hôpital, personnel soignant
- **Actionneurs:** moniteur pour afficher des questions, les résultats de tests ou de diagnostique, le traitement, etc.
- **Senseurs:** clavier et souris pour saisir les symptômes, les réponses aux questions, etc.

Types d'environnements

- **Complètement observables** (vs. partiellement observable): grâce à ses senseurs, l'agent a accès à l'état complet de l'environnement à chaque instant
- **Déterministe** (vs. stochastique): l'état suivant de l'environnement est entièrement déterminé par l'état courant et l'action effectuée par l'agent
- **Épisodique** (vs. séquentiel): les opérations/comportements de l'agent sont divisés en épisodes:
 - ◆ chaque épisode consiste à observer l'environnement et effectuer une seule action
 - ◆ et le choix de chaque action dans une épisode ne dépend que de cette épisode
- **Statique** (vs. dynamique): l'environnement ne change pas lorsque l'agent n'agit pas
- **Discret** (vs. continu): un nombre limité et clairement distincts de données sensoriels et d'actions
- **Agent unique** (vs. multi-agent): un agent opérant seul dans un environnement

Exemples d'environnements

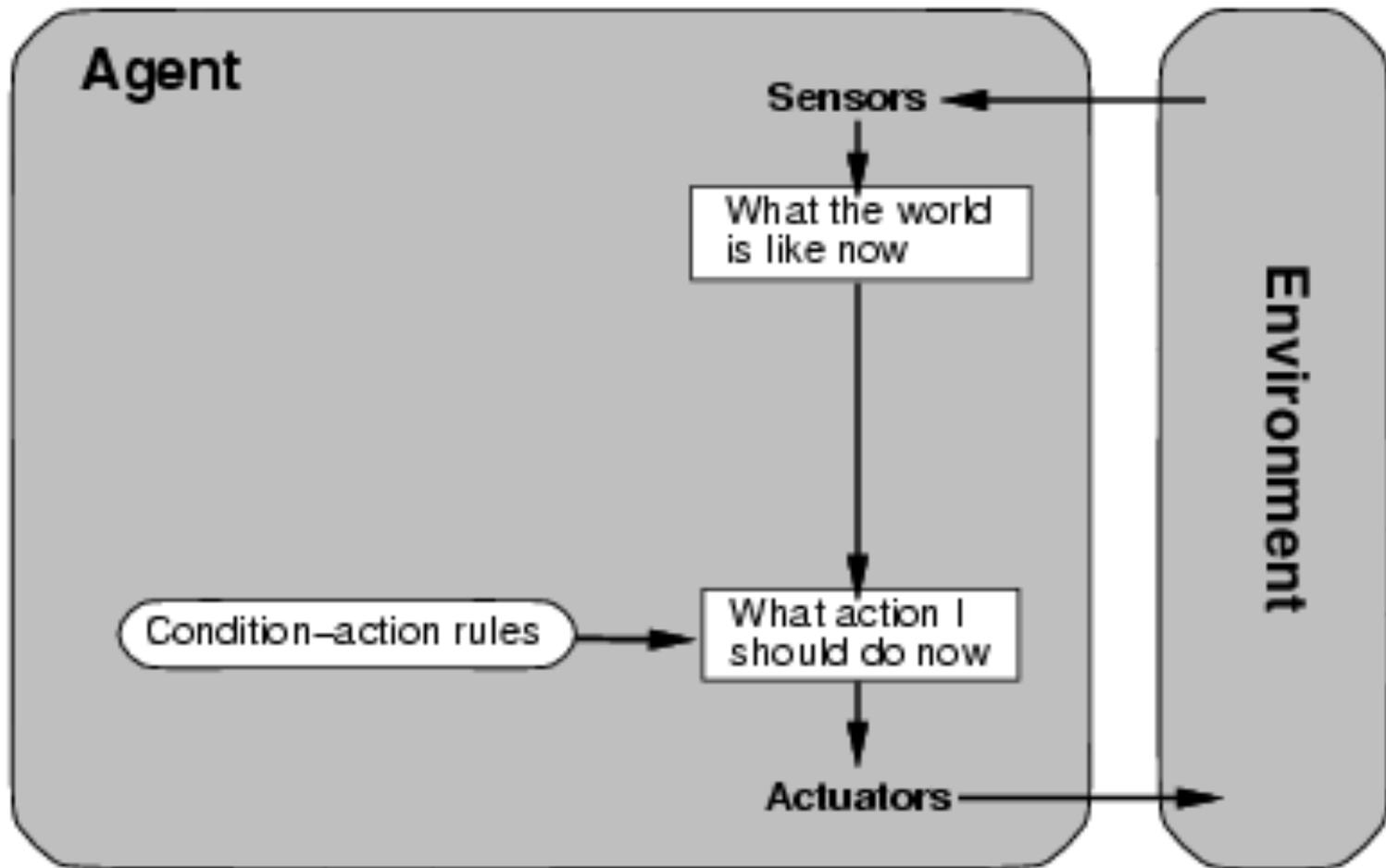
Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

Structure des agents

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

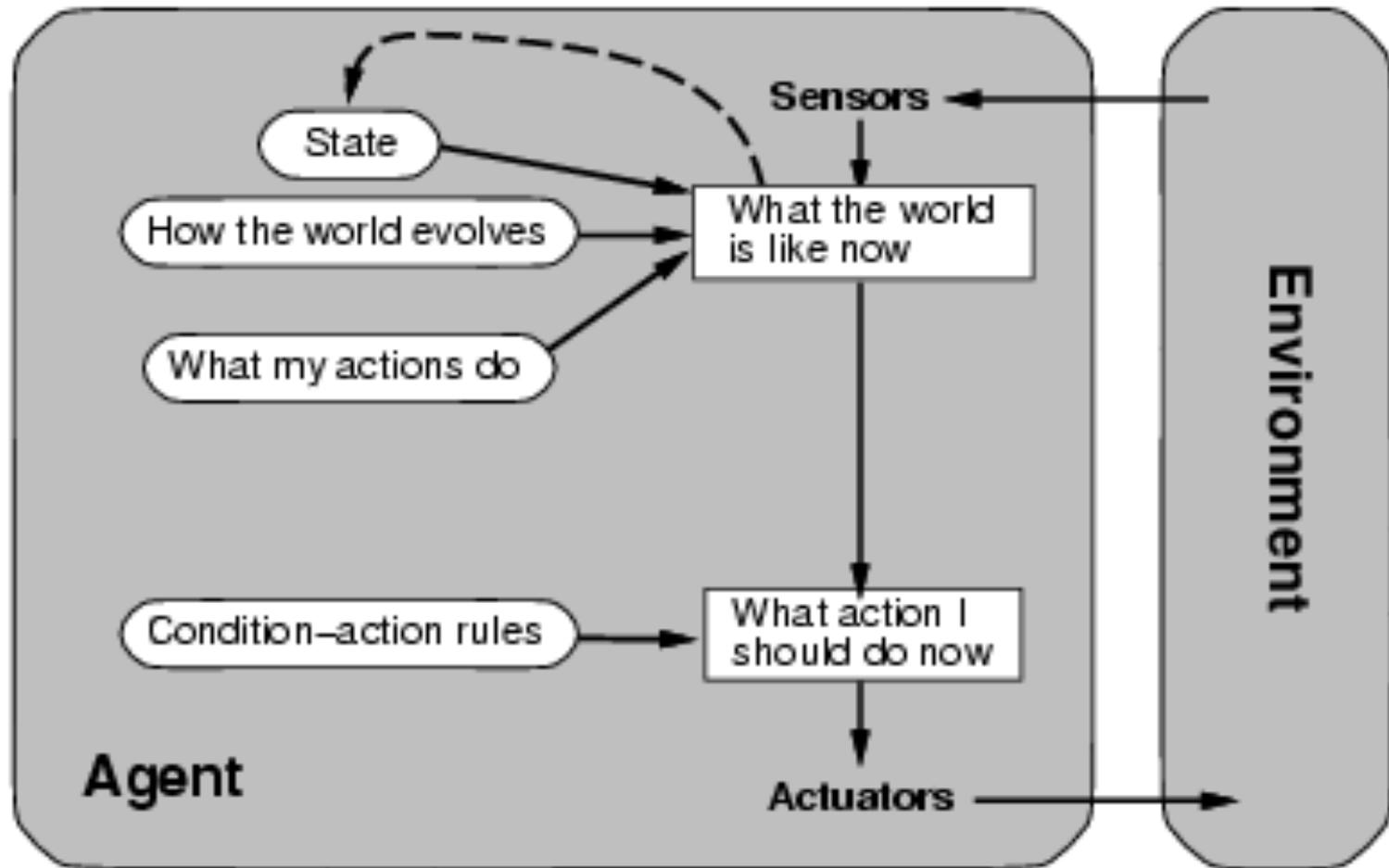
Simple reflex agents



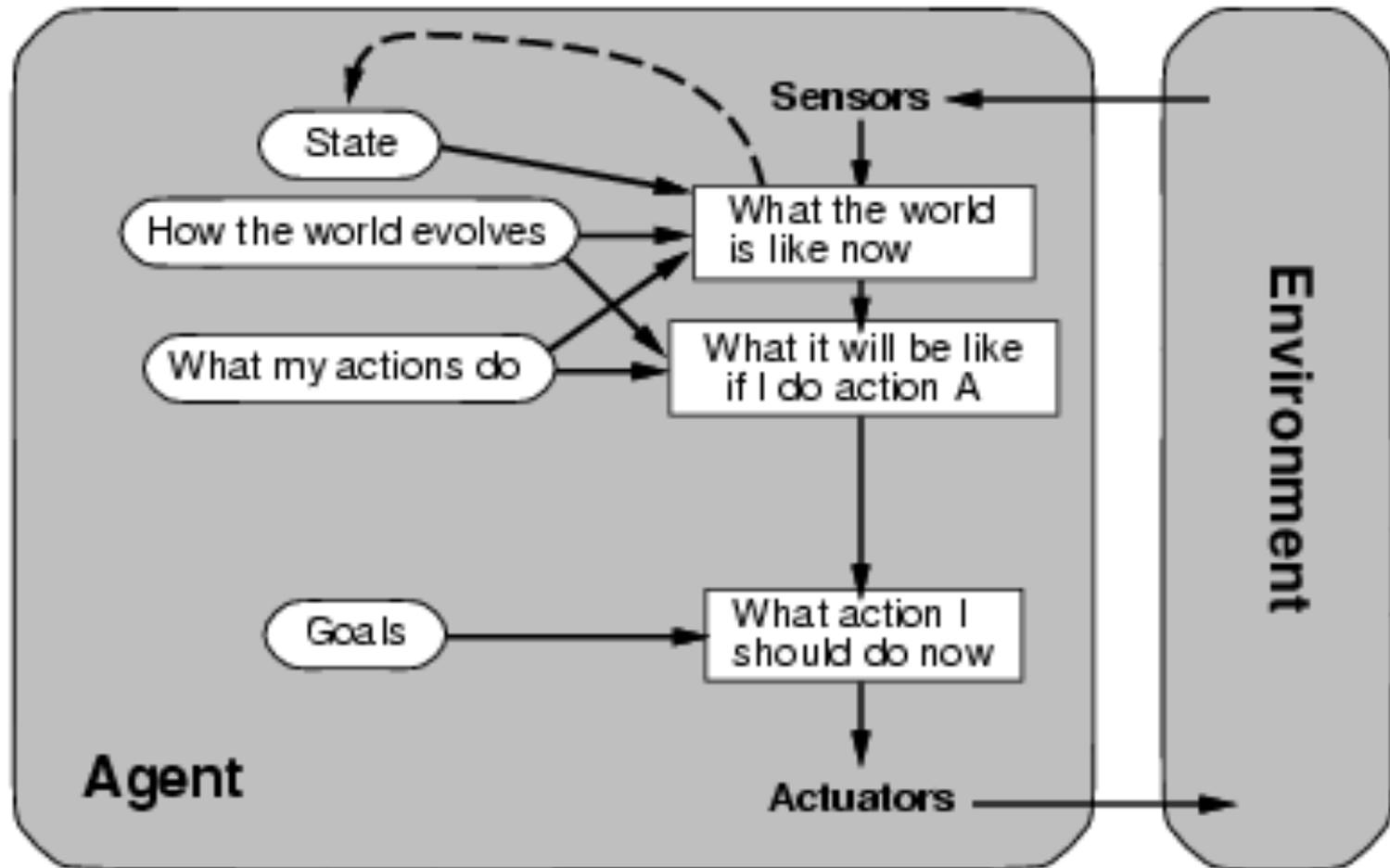
Simple reflex agents

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

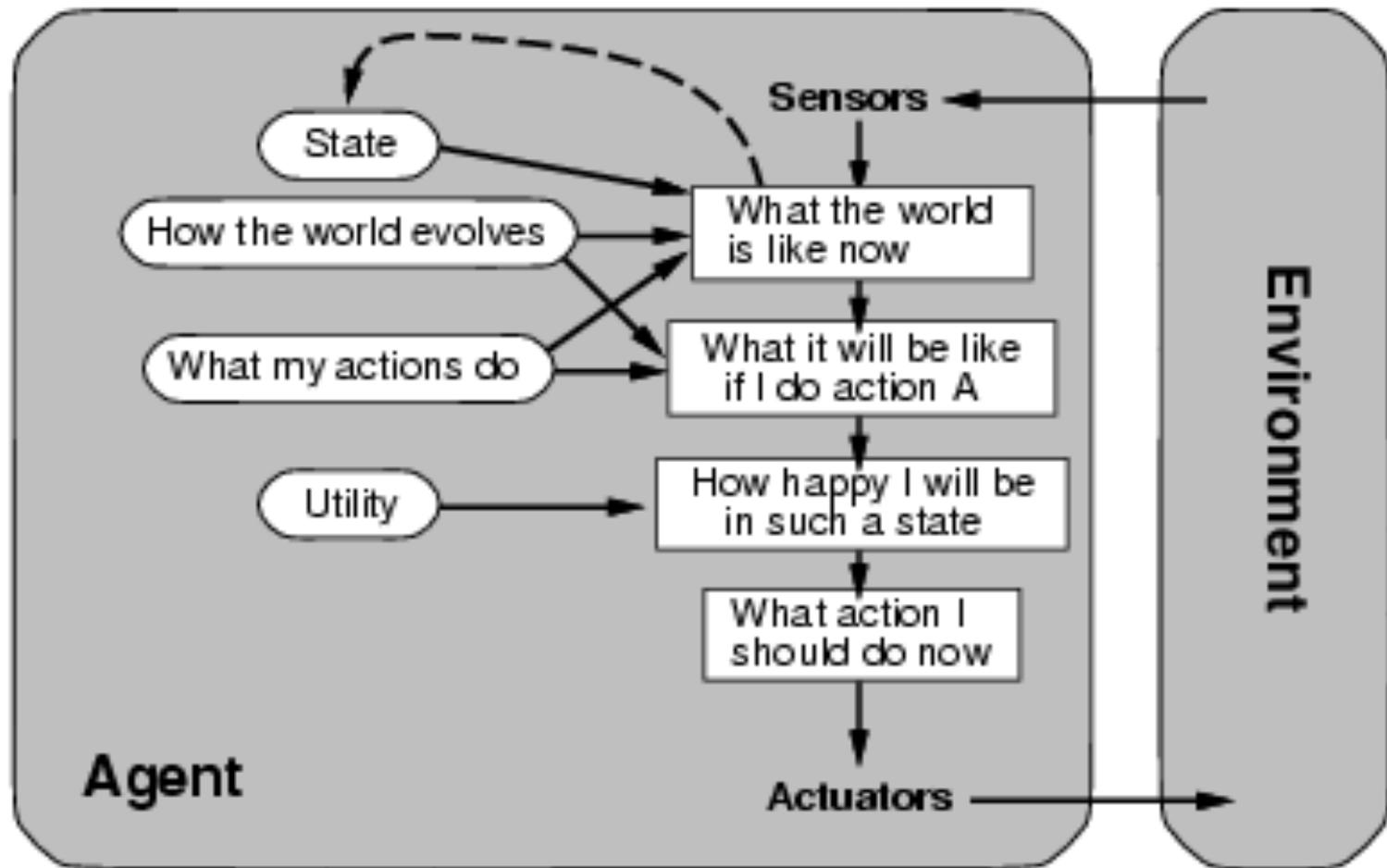
Model-based reflex agents



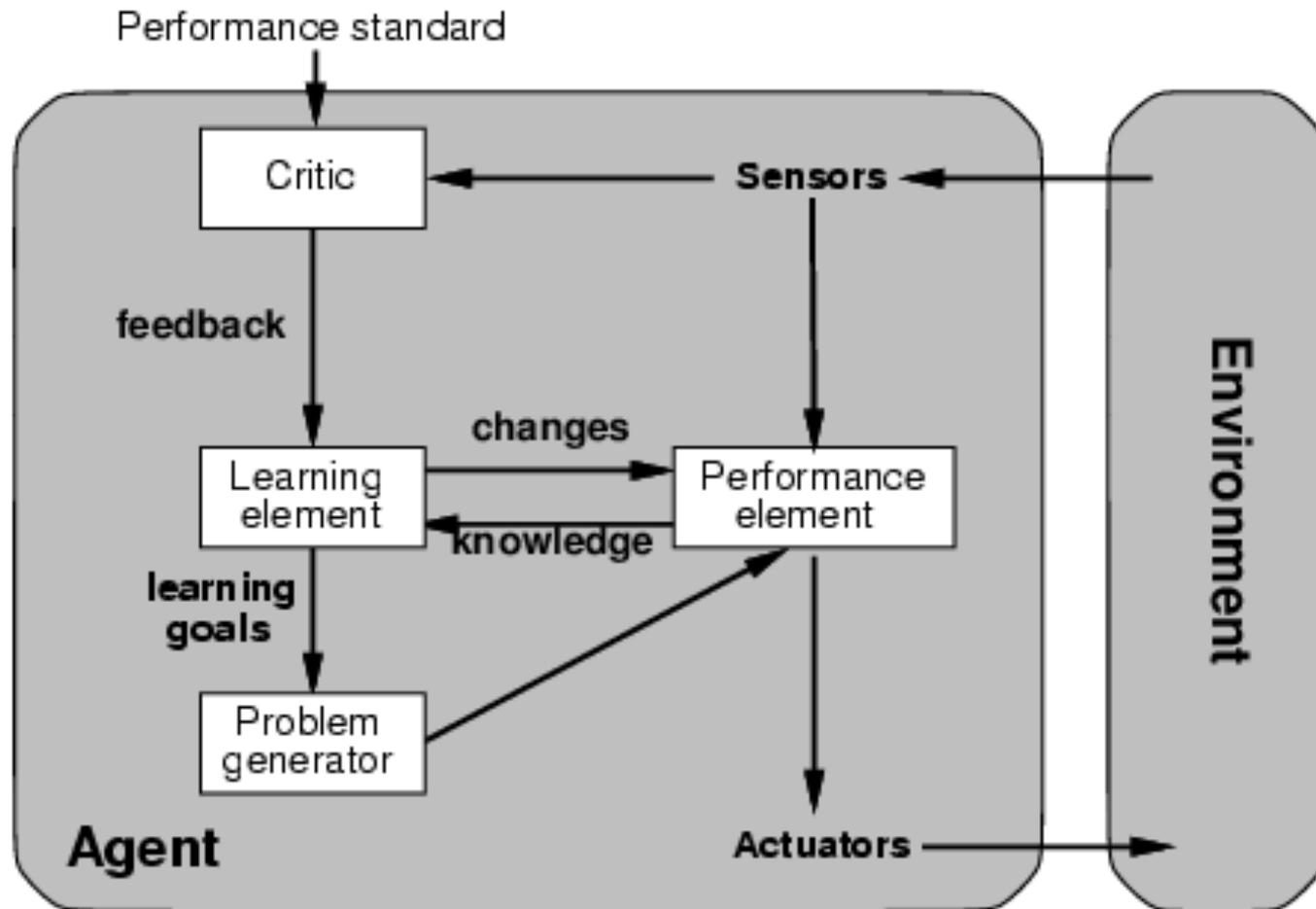
Goal-based agents



Utility-based agents

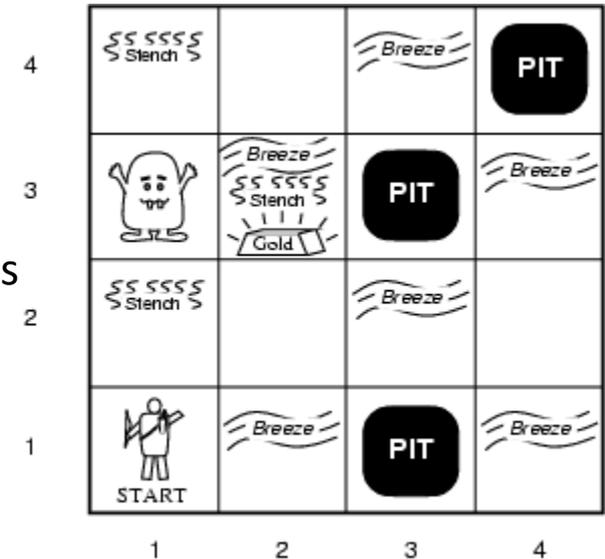


Learning agents



Exemple : le monde des wumpus (Section 7.2)

- Mesure de performance
 - ◆ or +1000, mort -1000
 - ◆ -1 par pas, -10 pour une flèche
- Environnement
 - ◆ puanteur dans les chambres adjacentes au wumpus
 - ◆ brise dans les chambres adjacentes à une fosse
 - ◆ scintillement si l'or est dans la chambre
 - ◆ le wumpus meurt si on lui tire une flèche de face
 - ◆ on a une seule flèche
 - ◆ on peut ramasser l'or dans la même chambre
 - ◆ on sort de la grotte en grimpant à la case [1,1]
- Senseurs: *Stench* (puanteur), *Breeze* (brise), *Glitter* (scintillement), *Bump* (choc), *Scream* (cri).
- Actionneurs: Left turn, Right turn, Forward, Grab, Climb, Shoot



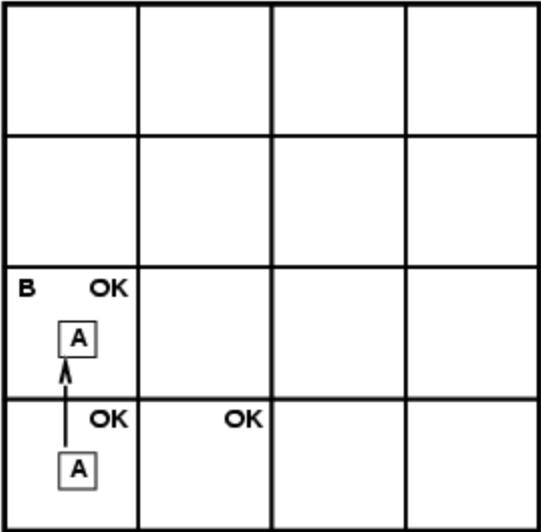
Caractéristiques du monde des wumpus

- Complètement observable? Non – seulement perception locale.
- Déterministe? Oui – l'effet de chaque action est prévisible.
- Épisodique? Non – séquentiel au niveau des actions.
- Statique? Oui – le wumpus et les fosses ne bougent pas.
- Discret? Oui.
- Agent unique? Oui – La seule action du wumpus est de nous « bouffer » si on atteint sa chambre.

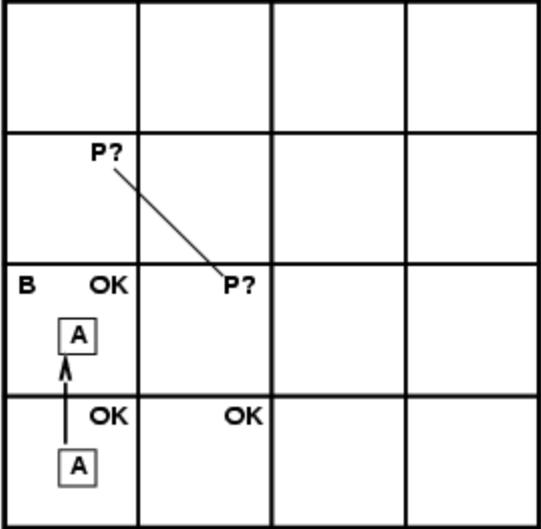
Exploration du monde des wumpus

OK			
OK A	OK		

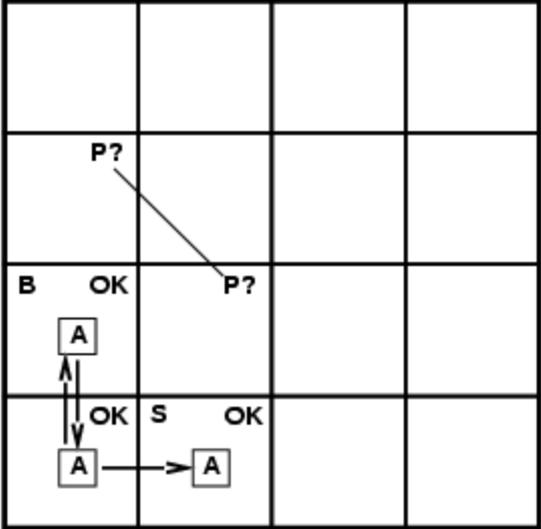
Exploration du monde des wumpus



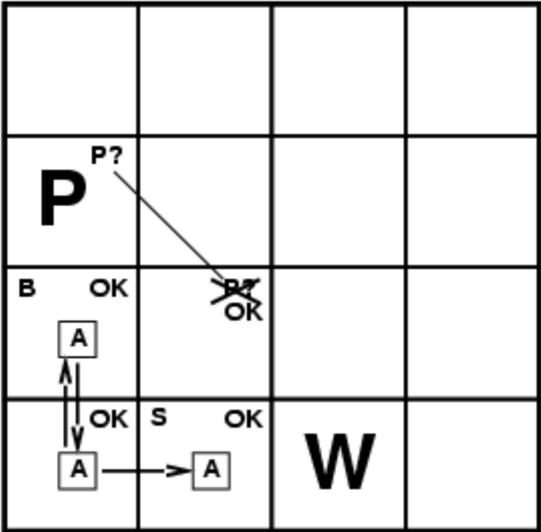
Exploration du monde des wumpus



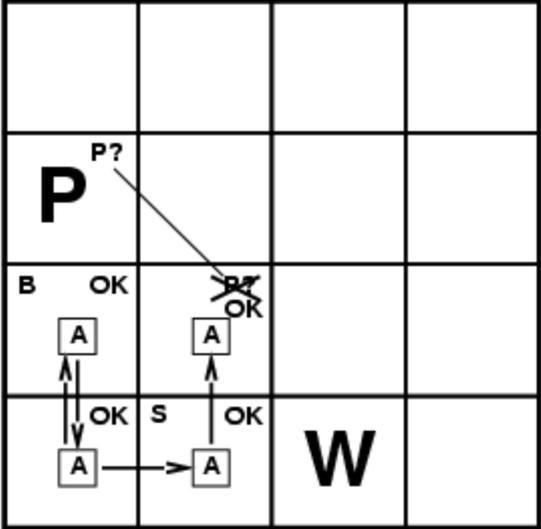
Exploration du monde des wumpus



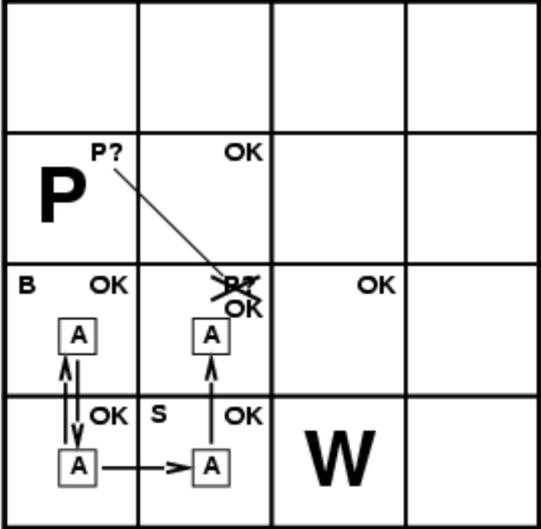
Exploration du monde des wumpus



Exploration du monde des wumpus



Exploration du monde des wumpus



Exploration du monde des wumpus

