

Introduction à l'intelligence artificielle

Rym Guibadj, Fabien Teytaud

LISIC, ULCO, EILCO

Contenu

- Agents logiques (raisonnement, déduction)
- Résolution de problèmes
- Optimisation

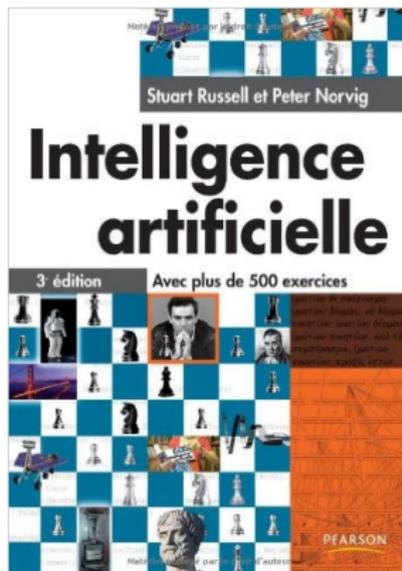
Volume horaire

- 7 x 2h Cours
- 6 x 2h TD / TD machine
- 3 x 4h TP

Evaluation

- TP
- Examen
- Note module = $0.3 \times \text{TP} + 0.7 \times \text{Examen}$

Référence



- Intelligence Artificielle, Stuart Russel et Perter Norving, edition Pearson
- Notes du cours de Fabien Tytaud (Université du Littoral Côte d'Opale)

Qu'est ce que l'intelligence artificielle ?

Quelques exemples de tâches "intelligentes" :

- le raisonnement de bon sens
- l'étude des sciences (physique, mathématiques, ...)
- la compréhension d'une langue
- l'écriture de logiciels
- la conduite d'un véhicule
- ...

Qu'est ce que l'intelligence artificielle ?

Il est difficile de donner une définition précise :

- la difficulté à définir l'Intelligence
- de nombreux domaines sont concernés par l'IA

Qu'est ce que l'intelligence artificielle ?

Une définition approximative

Reproduire le comportement humain

Définition de Marvin Lee Minsky

La construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains

Qu'est ce que l'intelligence artificielle ?

Beaucoup de définitions (et autant de voies de recherche)

- **L'approche cognitive** : l'IA est la réalisation des programmes imitant dans leur fonctionnement l'esprit humain. Exemple : la compréhension du langage naturel
- **L'approche pragmatiste** : une IA est une boîte noire intelligente qui réussit un certain nombre de tests. Exemple : Test de Turing
- **L'approche connexionniste** : une IA doit être capable d'apprendre à partir d'exemples et de généraliser (empiriquement) cet apprentissage

Qu'est ce que l'intelligence artificielle ?

Une définition générale

L'IA vise à faire exécuter par l'ordinateur des tâches pour lesquelles l'Homme est actuellement meilleur que la machine.

Remarque :

Un problème dit d'IA aujourd'hui ne le sera peut-être plus demain !

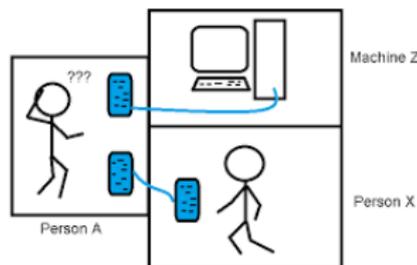
Origine de l'IA

- Des idées anciennes
 - Servantes d'Héphaïstos
 - Frankenstein
 - Le joueur d'échec de Maelzel
 - ...
- Naissance
 - Alan Turing "Computing Machinery and intelligence", 1950
 - McCarthy : Conférence au Dartmouth College, 1956

Origine de l'IA (1945-1955)

Test de Turing :

- Confrontation verbale d'un humain A avec une machine Z et un autre humain X à l'aveugle.
- A doit deviner qui est un homme et qui est une machine
- A pose des questions à X et Z



La machine réussira le test de turing si A n'est pas capable de dire lequel de ses interlocuteurs est une machine

Les balbutiements (1955-1970)

- 1956 :
 - John McCarthy invente le terme IA.
 - Postulat : toute activité intelligente est modélisable et reproductible par une machine.
 - LOGIC THEORIST : premier programme pour la démonstration de théorèmes
- 1957 :
 - GPS (General Problem Solver)
 - NSS (jeu d'échecs)
- 1960 : Algorithme de Davis et Putnam
- ...

Les balbutiements (1955-1970)

Premières Déceptions

Problème étudié : Traduction automatique \Rightarrow Résultat : un échec

Mais, on en tira des enseignements :

- importance des connaissances non exprimées
- étude de la représentation des connaissances
- impossibilité de représenter toutes les connaissances
- nécessité de les rédiger sous forme générique

La spécialisation (1970-1980)

De nombreuses ramifications vont se créer

- compréhension du langage naturel
- démonstration automatique de théorèmes
- représentation des connaissances
- programmation des jeux
- résolution de problèmes
- ...

L'essor(1980- ? ? ? ?)

Les domaines de l'IA :

- Jeux (Echecs, Quake, Wow, Stratcraft ...)
- Planification
- Système à base de connaissances
- Traduction automatique
- Dignostique médical
- Navigation autonome (avions, drones, robots, voitures...)
- Fouille de donnés
- Reconnaissance de formes
- Identification vocale ou visuelle
- Optimisation de processus industriels
- Interfaces intelligentes
- ...

Quelques succès de l'IA

- Deep blue [Jeu d'échec]
- Mogo [Jeu de Go]
- ViaVoice [Reconnaissance vocale]
- Codage postal [Reconnaissance d'écriture]
- Waston [Système expert]
- Aibi et Asimo [Robots de compagnie]
- Robots assistants [Robots médicaux]
- Bigdog [Robots militaires]
- Google Driveless Car [Véhicule autonome]
- ...

Quelques difficultés rencontrées en IA

- Difficultés de modélisation :
 - les problèmes ne sont pas toujours parfaitement définis
 - certaines notions sont difficiles à exprimer :
 - possibilité, probabilité, préférence, . . .
- Difficultés de résolution :
 - difficultés de conception des algorithmes
 - espaces de recherche très vastes
 - problèmes de temps de réponse

Objectifs

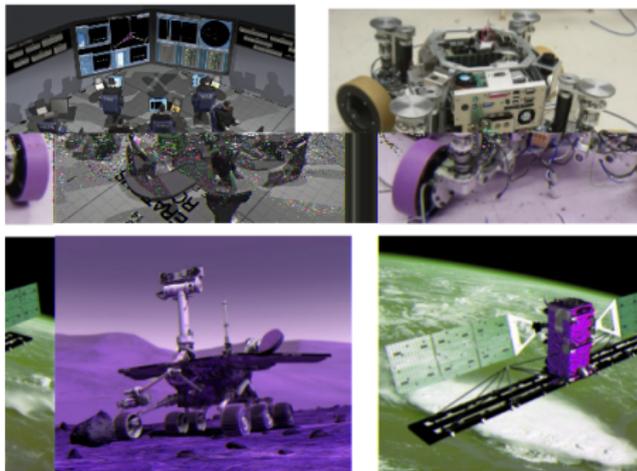
- Définir la notion d'agents intelligents (agent rationnel)
- Comprendre l'analyse PEAS (Performance measure, Environment, Actuators and Sensors)
- Distinguer divers types d'environnements
- Distinguer divers types d'agents

Création d'agents intelligents

Capacités fondamentales :

- perception
- représentation des connaissances (modélisation)
- apprentissage
- raisonnement
- prise de décisions

Exemples d'agents intelligents



- Systèmes d'aide à la décision
- Robots
- IA dans les jeux

Programmation d'actions vs décisions automatiques

- Programmation d'actions
 - scripts
 - machines à états finis
- Décision automatique
 - Les actions à exécuter ne sont ni scriptées ni programmées à l'avance
 - 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

Agents

Définition

Un agent est n'importe quelle entité qui perçoit son environnement par des **capteurs** (sensors) et qui agit sur cet environnement par des **actionneurs** (actuators)

- Un agent humain a :
 - des yeux, des oreilles ...etc.
 - des mains, des jambes, une bouche ...etc
- Un agent robot a :
 - des caméras, des capteurs infra rouges et autres capteurs
 - des roues, des jambes, des bras-articulés, et d'autres actionneurs
- Un agent logiciel a :
 - un clavier, un accès lecture à un disque dur ...etc.
 - un écran, un accès écriture à un disque dur ...etc.

Agents et environnements

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

$$f : p^* \rightarrow A$$

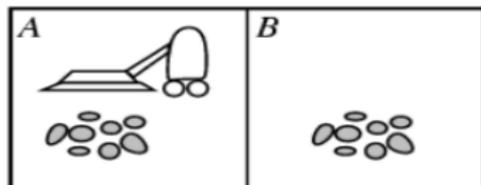
- Le processus est implémenté sous forme de programme sur une architecture matérielle particulière

Ebauche d'un agent

```
function AGENT(percept) returns action
static : memory

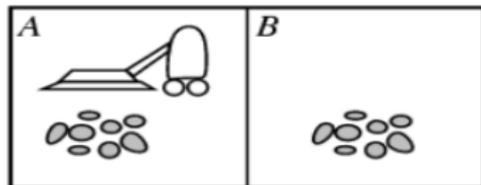
memory ← UPDATE-MEMORY(memory,percept)
action ← CHOOSE-BEST-ACTION(memory)
memory ← UPDATE-MEMORY(memory,action)
return action
```

Aspirateur robotisé



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

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 rationnels

- **Rationnel** \neq "**qui sait tout**" l'agent ne doit pas connaître tous les effets de ses actions
- **Rationnel** \neq "**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

Modèle PEAS

- PEAS : un modèle de conception des agents par la spécification des composantes majeures suivantes :
 - mesure de performance (Performance)
 - éléments de l'environnement (Environnement)
 - les actions que l'agent peut effectuer (Actionneurs ou Actuators)
 - la séquence des observations ou percepts de l'agent (Capteurs ou Sensors)
- PEAS : Performance, Environment, Actuators, Sensors

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, compteur de vitesse, GPS, témoins de moteurs, etc.

Modèle PEAS pour un diagnostic médical automatisé

- **Agent** : système de diagnostic 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 diagnostic, le traitement, etc.
- **Senseurs** : clavier et souris pour saisir les symptômes, les réponses aux questions, etc.

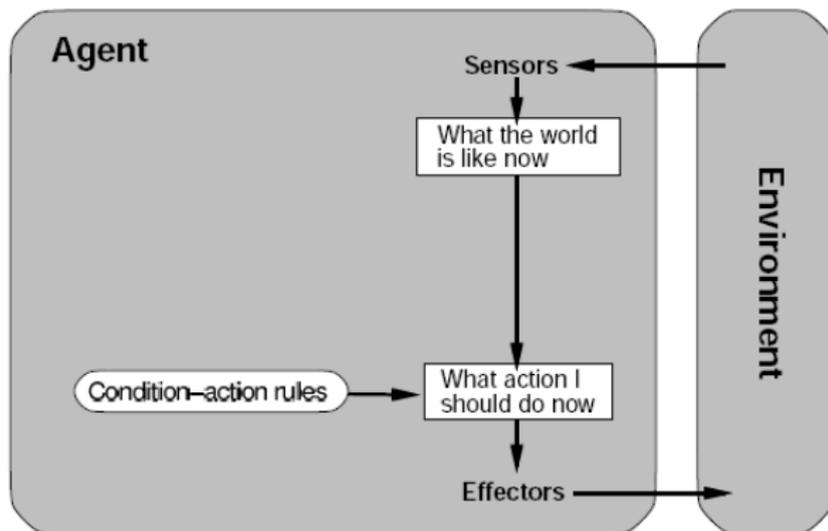
Caractéristiques d'environnement

- **Complètement observables** (vs partiellement observable)
- **Déterministe** (vs stochastique)
- **Episodique** (vs séquentiel)
- **Statique** (vs dynamique)
- **Discret** (vs continu)
- **Agent unique** (vs multi-agent)

Structure des agents

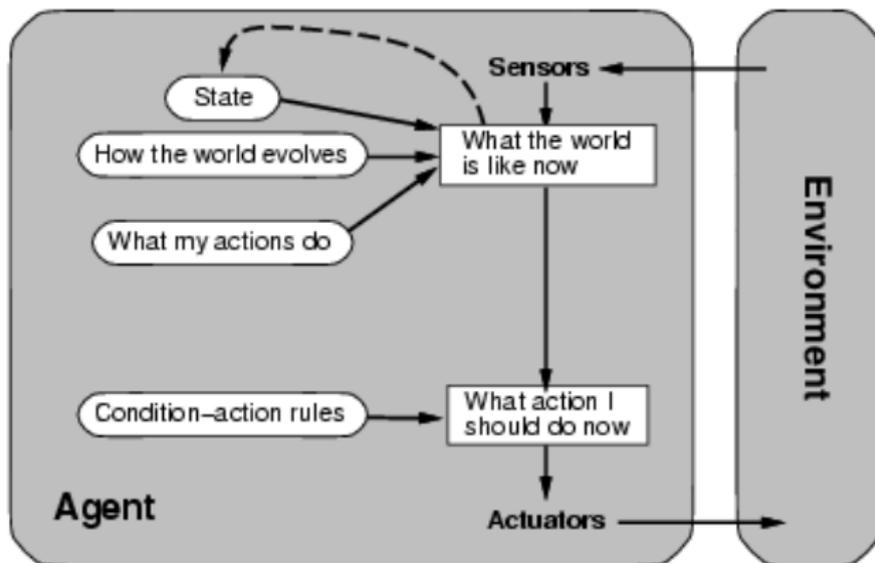
- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

Simple reflex agents



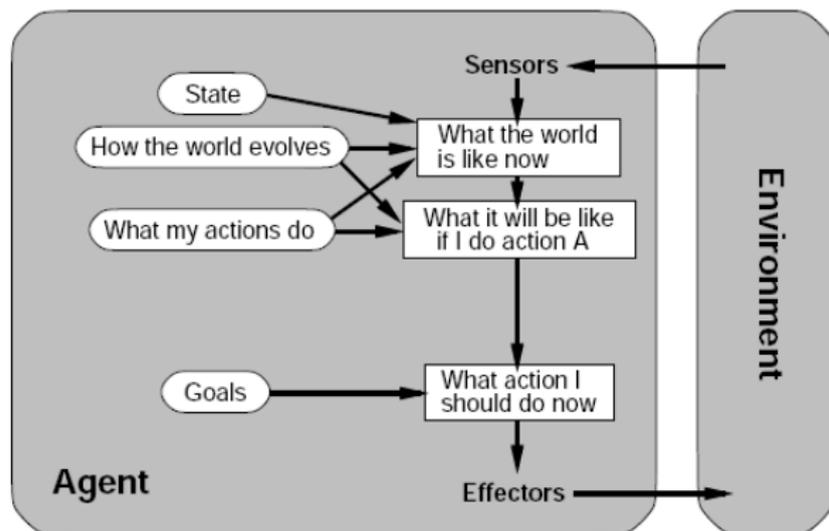
Agit seulement à partir du percept actuel en ignorant l'historique

Model-based reflex agents



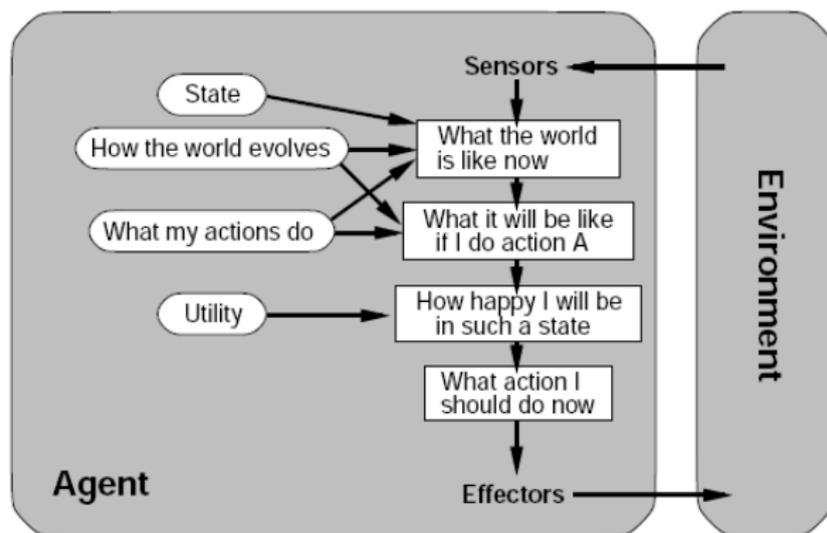
Accumule l'information dans le temps pour estimer l'état de l'environnement

Goal-based agents



Plutôt que de spécifier une règle conditions/ actions explicitement, on ne fait que spécifier un but (va pouvoir tenir compte du futur)

Utility-based agents



Intègre la notion de préférence entre les différentes actions. Par exemple : choisir l'action qui résout une tâche donnée le plus rapidement possible

Apprentissage dans un agent

- Les 4 types d'agents précédents varient dans la façon de prendre leur décision
- A partir de quelles connaissances prendre ces décisions ?
 - solution : **apprendre ces connaissances**
- On va voir plusieurs façons de faire l'apprentissage, et ce pour différents types d'agents