



INTELLIGENCE ARTIFICIELLE « I-ILIA-028 »

CHAPITRE 2 : AGENTS INTELLIGENTS ET SYSTÈMES MULTI-AGENTS

Pr. Sidi Ahmed Mahmoudi

sidi.mahmoudi@umons.ac.be

www.deepilia.com

12 février 2025

Introduction

I. Agents et environnements

II. Concept de rationalité

III. Environnements

IV. Systèmes multi-agents

V. Applications

Conclusion

C'est quoi un agent intelligent



Exemple d'agent intelligent : médecin virtuel

Autres
exemples



Introduction

- Place centrale **d'agents rationnels** pour **l'intelligence artificielle**
- Possibilité d'appliquer le concept de **rationalité** à une vaste gamme d'agents opérants dans des **environnements** imaginables
- **But :**
 - développer des principes destinés à la construction d'agents intelligents
 - Développer des systèmes qu'on peut raisonnablement quantifier d'intelligents

Introduction

I. Agents et environnements

II. Concept de rationalité

III. Environnements

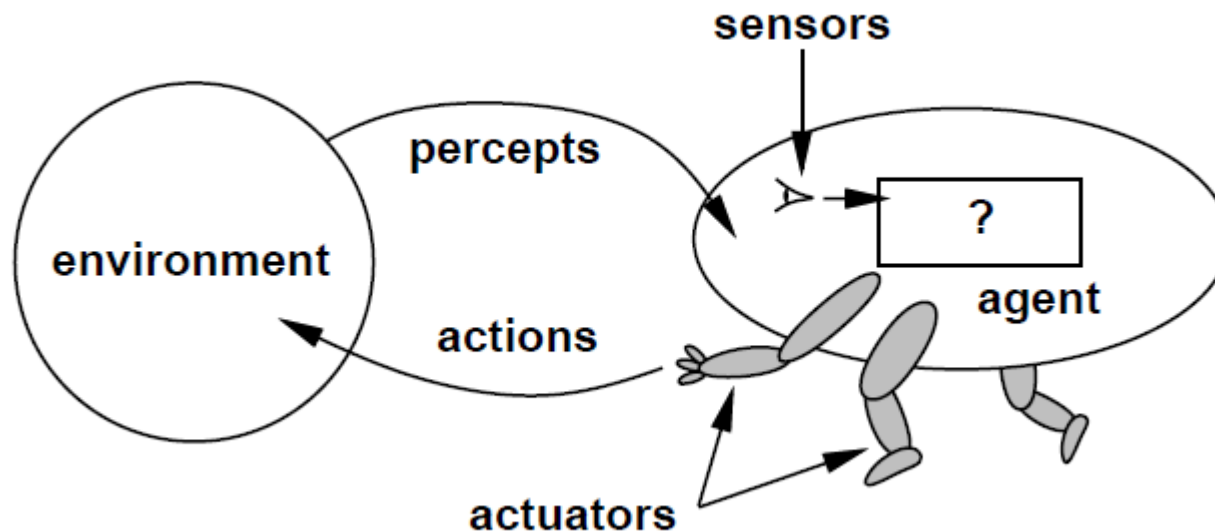
IV. Systèmes multi-agents

V. Applications

Conclusion

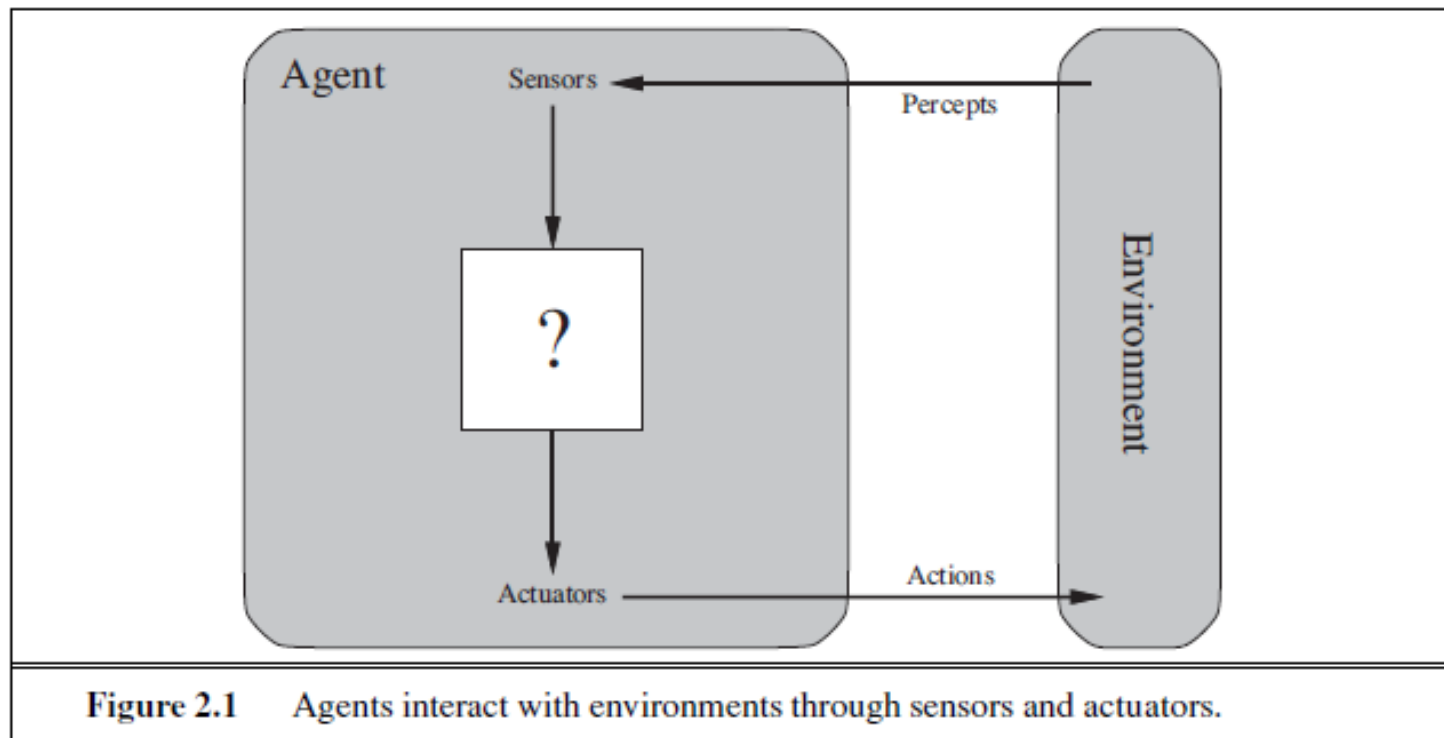
Agents et environnements

- **Agent** : toute entité pouvant **percevoir** son **environnement** grâce à des **capteurs** (senseurs) et qui agit sur cet environnement via des **effecteurs** (actuateurs)



Introduction

- Agent humain: capteurs (yeux, oreilles, etc.) et actuateurs (main, jambes, etc.)
- Agent robotique : capteurs (caméra, réseau, etc.) et actuateurs (fichiers, etc.)



Séquence de perception

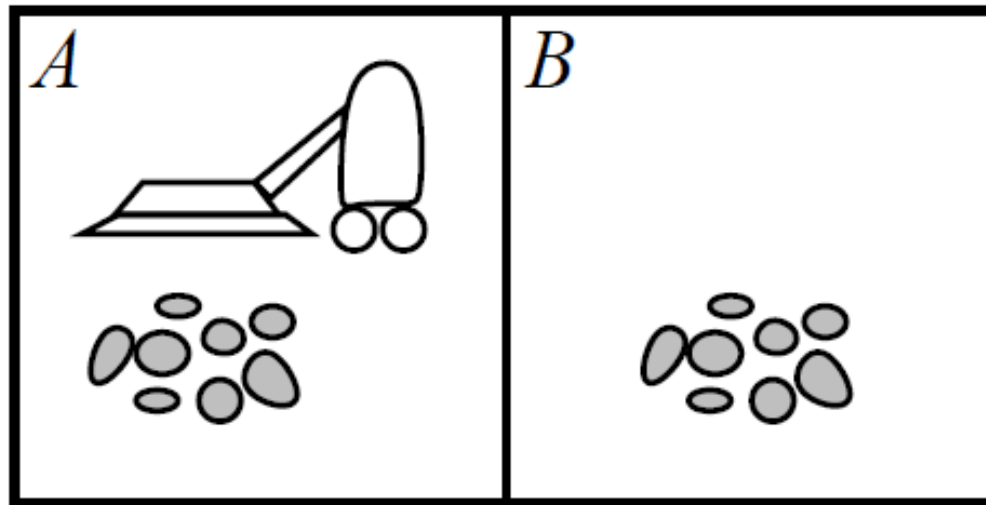
- Historique complet de tous ce que l'agent a perçu
- L'action choisie par un agent à un instant t dépend de la totalité des perceptions observées jusqu'à cet instant t
- Pas de prise en compte de ce qui n'a pas encore été perçu

Fonction d'agent

- La fonction d'agent permet de décrire le comportement de l'agent
- Cette fonction permet de correspondre une action à chaque séquence de perceptions
- Deux caractérisations possibles :
 - **Externe** : possibilité de tabuler la fonction d'agent afin d'associer pour chaque séquence de perceptions une action. Cette table est généralement de **très grande taille**. Elle peut même être infinie si on ne limite pas la longueur des séquences prises en compte
 - **Interne** : le programme d'agent est utilisé pour implémenter la fonction d'agent.

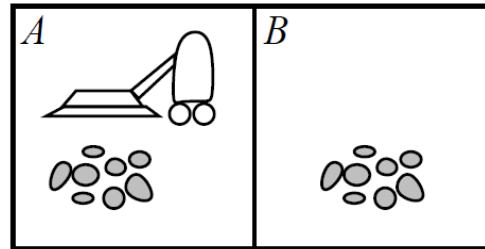
Exemple : le monde de l'aspirateur

- **Environnement** : deux carrés A et B pouvant être soit propre ou sale
- **Perceptions** : la localisation et le contenu
- **Actions** : se déplacer à gauche ou à droite, aspirer, rien.



Monde d'agent aspirateur limité à deux emplacements

Fonction d'agent



Percept sequence	Action
$[A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Dirty}]$	<i>Suck</i>
$[B, \textit{Clean}]$	<i>Left</i>
$[B, \textit{Dirty}]$	<i>Suck</i>
$[A, \textit{Clean}], [A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Suck</i>
\vdots	\vdots
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Suck</i>
\vdots	\vdots

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world

Programme d'agent

Implémentation de la fonction d'agent

```
def program(percept):  
    location, status = percept  
    if status == "dirty":  
        return "suck"  
    elif location == "A":  
        return "right"  
    elif location == "B":  
        return "left"
```

L'agent et son environnement

```
environment = Environment()  
agent = Agent(program)  
  
while True:  
    percept = environment.percept()  
    action = agent.program(percept)  
    environment.execute(action)
```

Fonction d'agent vs Programme d'agent

- **Fonction d'agent** : description mathématique abstraite

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

- **Programme d'agent** : implémentation concrète qui s'exécute sur un certain système physique pour produire la fonction f .

Introduction

I. Agents et environnements

II. Concept de rationalité

III. Environnements

IV. Systèmes multi-agents

V. Applications

Conclusion

Concept de rationalité

- Un agent rationnel est un agent qui effectue toujours l'action appropriée
- Toutes les entrées de la table de la fonction d'agent sont remplies correctement
- Comment déterminer qu'une action est bien appropriée ?
- Chaque agent propulsé dans un environnement génère une séquence d'actions et d'états

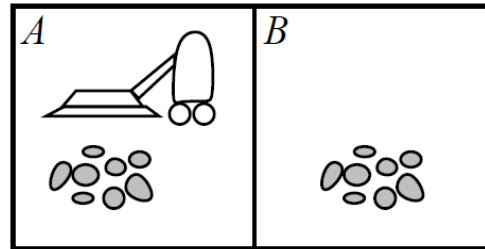
Concept de rationalité

- Mesure de performance : évaluer toute la séquence d'états de l'environnement (**et non pas de l'agent**).
- Le succès de l'agent ne doit pas être jugé par lui-même
- La mesure doit être faite du point de vue des conséquences de son comportement
- **Exemple** : un agent humain disant qu'il ne voulait pas une récompense (prix Nobel, Ballon d'Or, etc.) parce qu'il ne l'a pas obtenu.

Mesure de performances

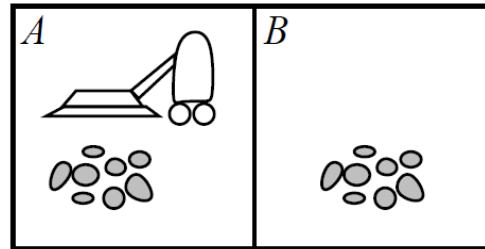
- Pas de mesure unique pour toutes les tâches et tous les agents
- Plusieurs façons et méthodes de mesure
- La mesure doit être bien appropriée au contexte
- La tâche de mesure de performances n'est pas aussi simple

Exemple du robot aspirateur



- Quelle est la bonne fonction d'agent? Comment formuler l'objectif de l'agent d'aspirateur?
 - ✓ 1 point par carré nettoyé à l'instant t ?
 - ✓ 1 point par carré propre par pas de temps, moins un par mouvement?
 - ✓ pénaliser pour un nombre $> k$ carrés sales?
- Peut-il être implémenté dans un petit programme d'agent?

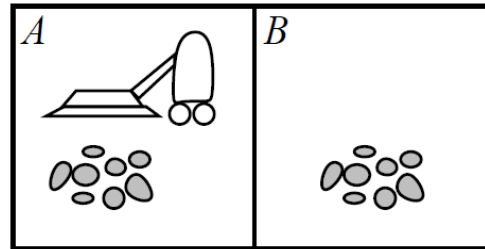
Exemple du robot aspirateur



- **Mesure de performance N° 01 :** mesure quantitative
 - ✓ Quantité de poussière ramassée au cours d'une journée de 8h
- **Mesure de performance N° 02 :** système de récompense/pénalité
 - ✓ 01 point pour chaque carré propre par unité de temps
 - ✓ Pénalité pour l'électricité consommée ou le bruit généré

Quelle mesure choisiriez-vous ?

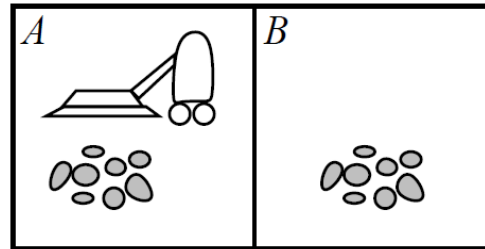
Exemple du robot aspirateur



- **Mesure de performance N° 01 :** mesure quantitative
 - ✓ L'agent rationnel fournit ce qu'on lui demande : il peut maximiser cette mesure en aspirant la poussière, puis la déverser sur le sol, en la ramassant de nouveau et ainsi de suite

Il vaut mieux concevoir les mesures en fonction de ce qu'on souhaite obtenir dans l'environnement plutôt que de penser au comportement de l'agent.

Exemple du robot aspirateur



- **Mesure de performance N° 02 :** système récompense pénalité
 - ✓ Meilleure évaluation car elle évalue la propreté de l'environnement (souhait réel)
 - ✓ La aussi, il peut y avoir deux types de comportements d'agents :
 - Travail médiocre et moins rapide
 - Travail énergique et rapide mais en prenant plusieurs pauses

Encore un choix à faire ?

Facteurs de rationalité

- La rationalité à un moment t dépend de quatre facteurs :
 - **Mesure de performance** : définissant les critères de succès
 - **Connaissance de l'environnement**: précédemment acquise par l'agent
 - **Les actions** : que l'agent peut réaliser
 - **La séquence de perception**: de l'agent au moment considéré
- Définition de la rationalité:

« pour chaque séquence de perception possible, un agent rationnel doit sélectionner une action, qui doit maximiser sa mesure de performance

Pour cela, il faut connaître la séquence de perceptions ainsi que la connaissance interne de l'agent »

Question ?

- Revenons à la fonction d'agent du robot aspirateur.

```
def program(percept):  
    location, status = percept  
    if status == "dirty":  
        return "suck"  
    elif location == "A":  
        return "right"  
    elif location == "B":  
        return "left"
```

- S'agit-il d'un agent rationnel ?

Réponse ?

- Pour répondre à cette question, il faut préciser la mesure de performance, la connaissance de l'environnement, les capteurs et effecteurs de l'agent.

Posons les hypothèses suivantes :

- Un point accordé pour chaque case nettoyée sur une durée de 1000 unités t
- La géographie de l'environnement est connue à priori (figure 11)
- Distribution et emplacement initial de la poussière non connus à priori
- Une case propre reste propre
- Actions possibles : Gauche, Droite, Aspirer
- L'agent reste en place si une action mène vers l'extérieur d'environnement
- L'agent perçoit sa localisation et sait si elle contient de la saleté

Petit exercice ?

- Prenons les hypothèses décrites précédemment
- Considérons un nombre n de carrés sales et 1000 unités de temps
- Sachant que 1000 unités de temps sont suffisantes pour tous nettoyer, quel serait la mesure de performance de l'agent ?
- Cette mesure risque t-elle de changer?
- Si on donne une pénalité pour chaque mouvement quel serait le score ?
- Comment peut-on améliorer cet agent ?

Omniscience, apprentissage et autonomie

- **Agent omniscient** : connaît le résultat réel de ses actions et peut agir en conséquence
- La rationalité n'est pas synonyme de perfection
- Rationalité : maximise la performance espérée
- Perfection : maximise la performance réelle
- Rationalité \neq Omniscience
- La rationalité n'implique pas l'omniscience car un choix rationnel ne dépend que de la séquence de percepts à un moment déterminé
- Il faudra tout de même s'engager à ce que l'agent n'engage pas des activités stupides

Récolte d'informations et apprentissage

- **Récolte d'informations** : réaliser une action dans le but de modifier les futurs percepts (information gathering)
- **Exemple** : regarder avant de traverser la rue
- **Apprentissage** : agent acquiert de l'expérience et se retrouve dans des situations dans lesquels l'environnement est connu à priori.
- L'agent n'a plus besoin de collecter de l'information
- **Exemple** : conduire sans utiliser le GPS

Autonomie

- **Manque d'autonomie** : un agent dépend plus des connaissances préalables fournies par son concepteur que ses propres percepts
- **Agent autonome**: agent qui apprend pour compenser ses connaissances à priori partielles ou incorrectes
- **Exemple** : un agent aspirateur qui apprend à prévoir où et quand apparaîtra la poussière se comportera mieux qu'un agent qui ne le fait pas

La récolte d'informations, apprentissage et autonomie représentent des composantes principales de la rationalité

Introduction

I. Agents et environnements

II. Concept de rationalité

III. Environnements

IV. Systèmes multi-agents

V. Applications

Conclusion

Environnements

- Pour construire un agent rationnel, il faut réfléchir aux environnements et tâches : définition d'un environnement de travail
- Cet environnement est défini par 04 éléments :
 - Mesure de **P**erformance
 - **E**nvironnement
 - **A**ctionneurs
 - **S**enseurs
- Description **PEAS**

Exemple : Robot chauffeur de taxi

- Description de l'environnement PEAS:
 - Mesure de Performance
 - Environnement
 - Actionneurs
 - Senseurs
- Quel serait votre proposition PEAS ?

Exemple : Robot chauffeur de taxi

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.

Question ?

- Quel description PEAS proposez vous pour l'exemple d'agent d'aide d'achat en ligne ?

Types d'environnements

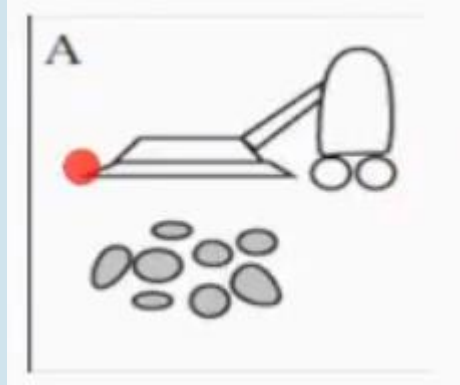
- Entièrement observable **vs** partiellement observable
- Mono-agent **vs** Multi-agent
- Déterministe **vs** stochastique
- Episodique **vs** séquentiel
- Statique **vs** dynamique
- Discret **vs** continu
- Connu **vs** inconnu

Entièrement observable vs partiellement observable

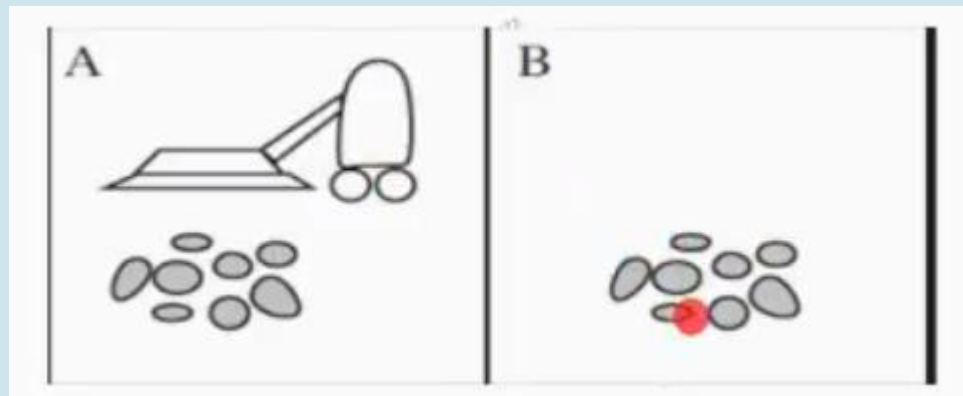
- **Entièrement observable:**
 - Capteurs d'agent donnant un accès complet à l'état d'environnement
 - Capteurs détectant tous les aspects pertinents pour le choix d'action
 - Pas besoin de garder une trace de l'environnement en interne
- **Partiellement observable:**
 - Bruits affectant les données transmises via les capteurs
- **Non observable:**
 - L'agent ne dispose d'aucun capteur

Entièrement observable vs partiellement observable

- Entièrement observable:

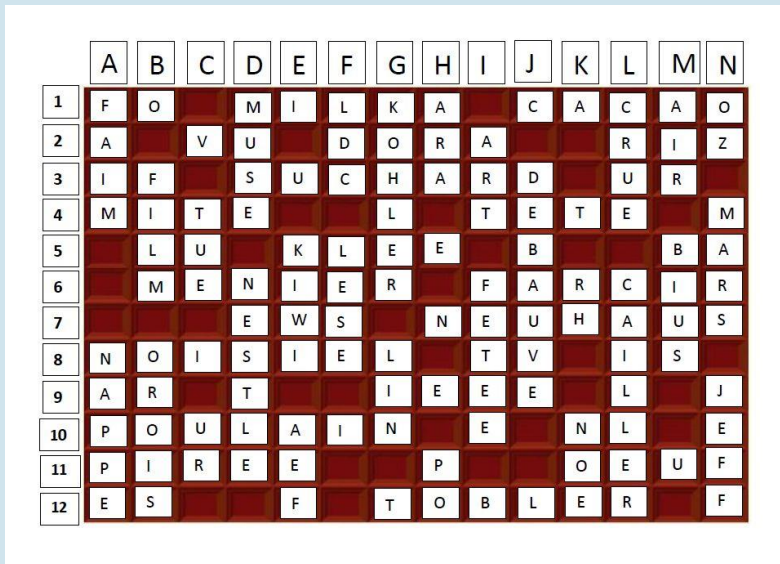


- Partiellement observable:



Mono-agent vs Multi-agent

Jeux mots croisés



Mono-agent

Jeux d'échecs

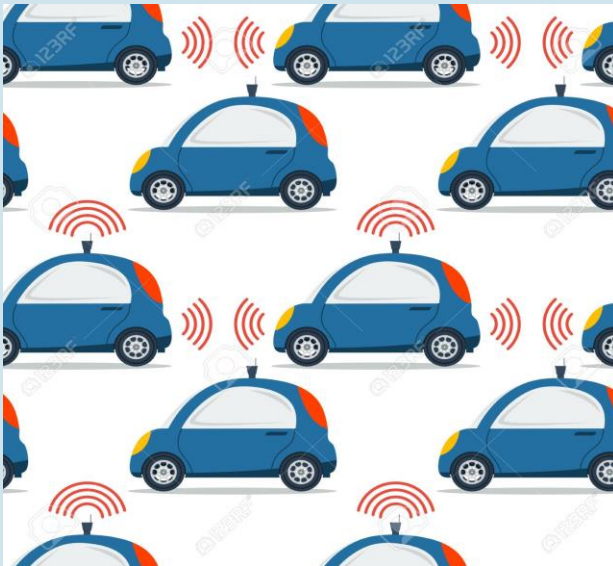


Multi-agent (2 agents)

Environnements Multi-agent

- On distingue des environnement multi-agents **compétitifs** et **coopératifs**

Robot chauffeur taxi



Coopératif

Multi-agent



Compétitif

- Le robot taxi peut être aussi partiellement compétitif (stationnement)

Déterministe vs Stochastique

- **Déterministe:**
 - L'état suivant de l'environnement est complètement déterminé par l'état courant et par l'action de l'agent
- **Stochastique:**
 - Aucune dépendance entre l'état courant (et action) avec l'état suivant de l'environnement
- **Incertain:**
 - Ni complètement observable, ni déterministe

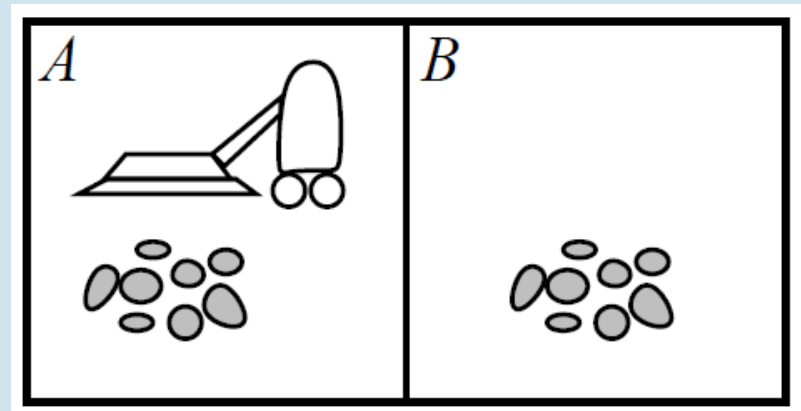
Déterministe vs stochastique

Rouler avec roue de secours



Stochastique

Robot aspirateur



Déterministe

Episodique vs séquentiel

- **Episodique:**
 - Expérience de l'agent divisée en épisodes atomiques
 - A chaque épisode, l'agent reçoit 1 percept et exécute 1 action unique
 - L'épisode suivant ne doit pas dépendre des actions précédentes
 - **Exemple :** l'agent détectant les pièces défectueuses sur une chaîne prend chaque décision indépendamment des précédentes. La décision courante n'influence pas sur la qualité de la pièce suivante.

Episodique vs séquentiel

- Séquentiel:
 - Décision actuelle susceptible d'affecter toutes les décisions futures
 - **Exemple** : Conduite automatique et jeux d'échecs

Dans les deux cas, les actions à court terme peuvent avoir des conséquences à long terme.

Environnements épisodiques sont plus simples que les environnement séquentiels, car les agents n'ont pas besoin d'envisager la suite

Statique vs dynamique

- **Dynamique:**
 - L'environnement peut changer pendant la fonction d'agent
 - L'agent doit toujours rester à l'écoute de tous changement possible
- **Statique:**
 - L'environnement ne peut pas changer pendant la fonction d'agent
 - Plus facile à gérer
- **Semi-dynamique:**
 - L'environnement ne change pas mais le score de l'agent change

Statique vs dynamique

Jeux mots croisés

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	F	O		M	I	L	K	A		C	A	C	A	O
2	A		V	U		D	O	R	A			R	I	Z
3	I	F		S	U	C	H	A	R	D		U	R	
4	M	I	T	E			L		T	E	T	E		M
5		L	U		K	L	E	E		B			B	A
6		M	E	N	I	E	R		F	A	R	C	I	R
7				E	W	S		N	E	U	H	A	U	S
8	N	O	I	S	I	E	L		T	V		I	S	
9	A	R		T			I	E	E	E		L		J
10	P	O	U	L	A	I	N		E		N	L		E
11	P	I	R	E	E			P			O	E	U	F
12	E	S			F		T	O	B	L	E	R		F

Statique

Chauffeur de taxi



Dynamique

Question ?



Statique ou dynamique ?

Discret vs continu

La distinction discret/continu s'applique à l'état de l'environnement, à la façon dont le temps est géré ou encore aux percepts et actions de l'agent

Exemple :

- Jeux d'échecs : **discret**, englobe un ensemble discret d'actions et percepts
- Taxi : continue, vitesse et position passent par des intervalles continus

Connu vs inconnu

- Cette distinction ne se rapporte pas à l'environnement mais à la connaissance que l'agent possède sur les lois de la physique qui régissent son environnement
- **Connu:**
 - Les issues de toutes les actions sont connues
 - Un environnement connu peut être partiellement observable
 - **Exemple** : jeu de solitaire, connaissance des règles mais pas de connaissance si les cartes seront retournées



Connu vs inconnu

- **Inconnu:**
 - L'agent doit apprendre comment marcher pour prendre les bonnes décisions
 - Un environnement inconnu peut être totalement observable
 - **Exemple :** Jeux vidéo, connaissance de l'intégralité de l'état du jeu via l'écran, mais pas de connaissance du rôle des boutons non essayés

Quelques exemples

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 et types d'agents

- Fonctionnement interne d'un agent
- **Objectif de l'IA :**
 - concevoir un programme agent qui implémente la fonction d'agent
 - associer des percepts à des actions
 - prévoir une architecture : équipement informatique doté de capteurs et d'effecteurs.

Agent = architecture + programme

- Le programme doit être approprié à l'architecture

Structure et types d'agents

- **Exemples d'architecture:**
 - Architecture dotée de jambes pour l'action marcher
 - PC, Voiture équipée d'ordinateur de bord, caméra, etc.
- **Rôle de l'architecture:**
 - Transmission des percepts (capteurs) au programme
 - Exécution du programme
 - Acheminement des actions générées vers les effectueurs.

Programme d'agent

- Un programme d'agent peut être implémenté via quatre méthodes :
 - Via des tables
 - Via des règles
 - Via des algorithmes de recherche
 - Via des algorithmes d'apprentissage

Agents pilotés par table

```
def TableDrivenAgentProgram(table):  
    percepts = []  
    def program(percept):  
        percepts.append(percept)  
        action = table[percepts]  
        return action  
    return program
```

- Un agent piloté par une table détermine son action via une table qui contient l'action appropriée pour chaque séquence de percept possible.
- **Problème de conception:** il faut anticiper toute la séquence des percepts et comment l'agent doit réagir.
- **Problème technique :** Table de recherche contenant $\sum_{t=1}^T |\mathcal{P}|^t$ entrées

Exemple : voiture autonome

- Pour une voiture autonome si on utilise la table:
 - Capteur (caméra): 30 fps avec une résolution de 640 x480
 - Durée : une heure
 - Nombre d'entrée : $10^{250000000000}$
 - Via **des algorithmes d'apprentissage**



Types de programmes d'agents

- Quatre types fondamentaux:
 - Agents réflexes simples
 - Agents réflexes fondés sur des modèles
 - Agents fondés sur des buts
 - Agents fondés sur l'utilité
 - Agents capables d'apprentissage

Introduction

I. Agents et environnements

II. Concept de rationalité

III. Environnements

IV. **Systèmes multi-agents**

V. Applications

Conclusion

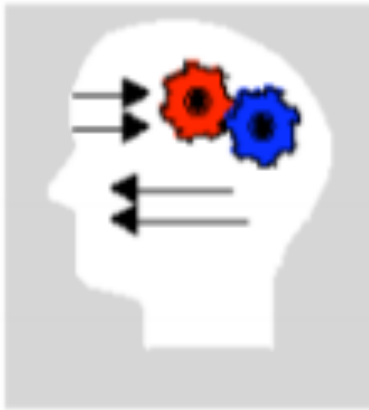
Systèmes multi-agents (SMA)

- Les Systèmes Multi Agents (SMA) se montrent très adaptés là où les outils classiques de modélisation ne sont pas suffisants
- Modéliser des phénomènes dans lesquels les interactions entre les différentes entités sont assez complexes.
- Les SMA permettent de représenter des entités autonomes (avec comportements) et gérer la coopération, négociation et communication avec d'autres entités.

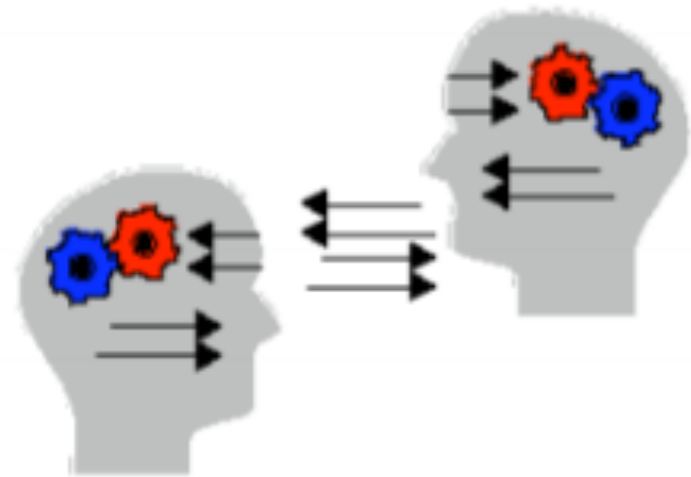
Systèmes multi-agents (SMA)

- **SMA:** un système dans lequel des agents artificiels opèrent collectivement et de façon décentralisée pour accomplir une tâche

SMA = Agents + Environnement + Interactions + Organisations (AEIO)



Penseur isolé



Communauté de penseurs

Systemes multi-agents (SMA)

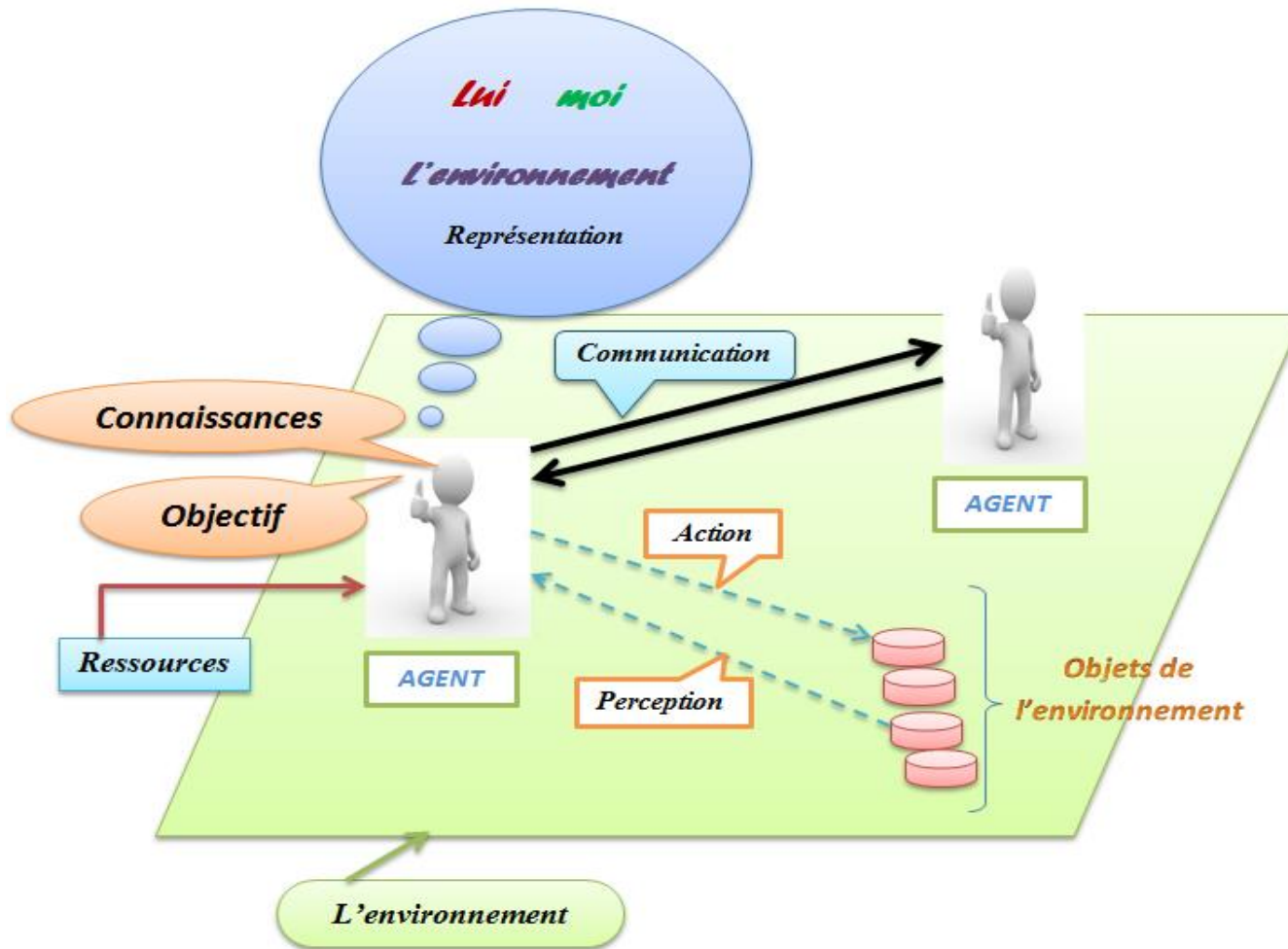


Agents

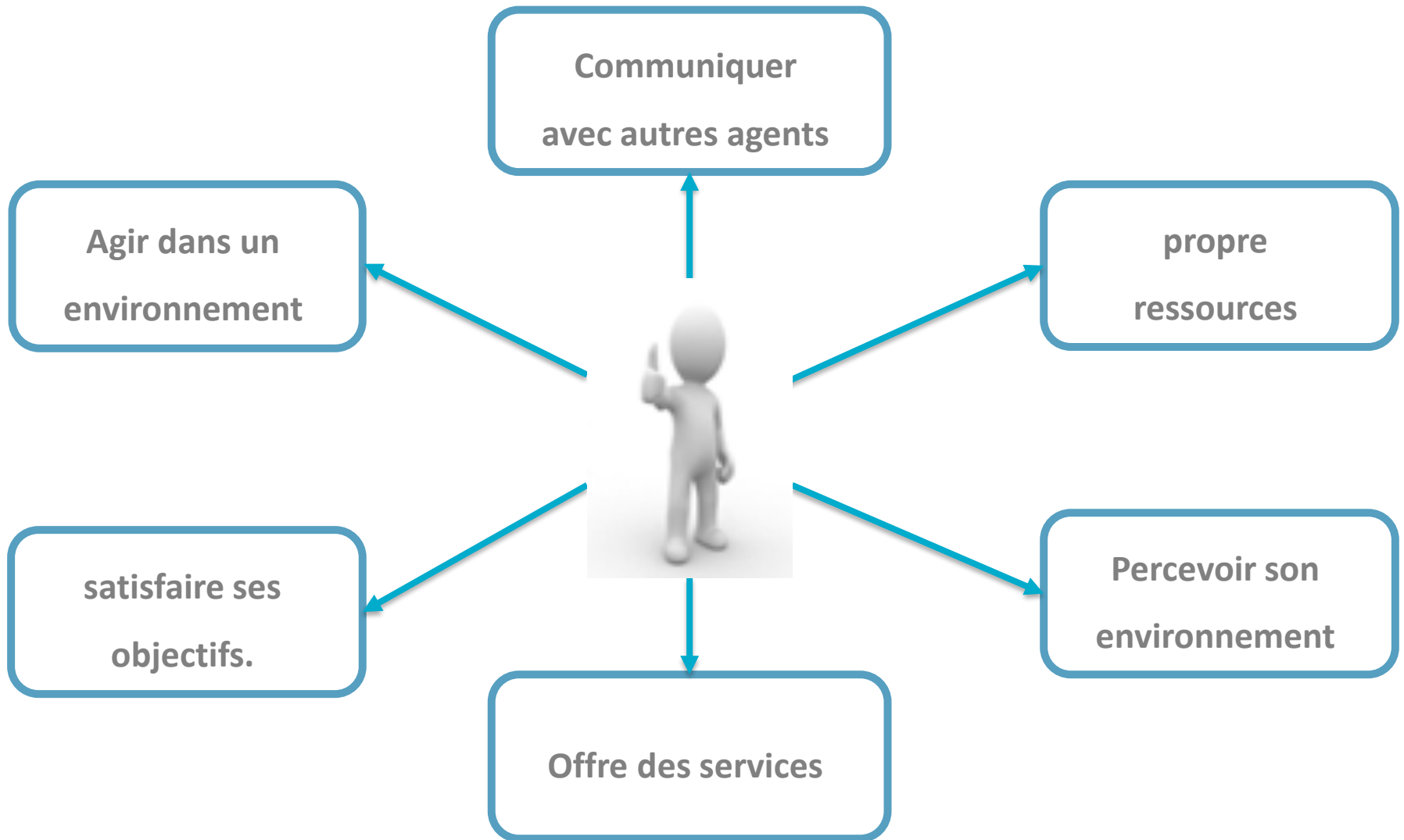
« Un agent est une entité réelle ou virtuelle :

- évoluant dans un environnement capable de le percevoir et d'agir dessus
- peut communiquer avec d'autres agents
- exhibe un comportement autonome, vu comme conséquence des interactions avec d'autres agents et buts poursuivis », **Jacques Ferber.**

Agents



Agents



Propriétés d'un d'agent

Autonome

contrôle ses actions sans intervention de l'utilisateur

Proactif

capacité de prendre l'initiative/comportement orienté but

Flexible

Capacité de répondre à temps

Social

capacité d'interagir avec d'autres agents ou utilisateurs

Situé

percevoir l'environnement, réagir en temps réel aux changements

Typologie des agents

Agent réactif

Il se contente de réagir aux changements dans l'environnement
Capture et réagit à des perceptions en appliquant des règles

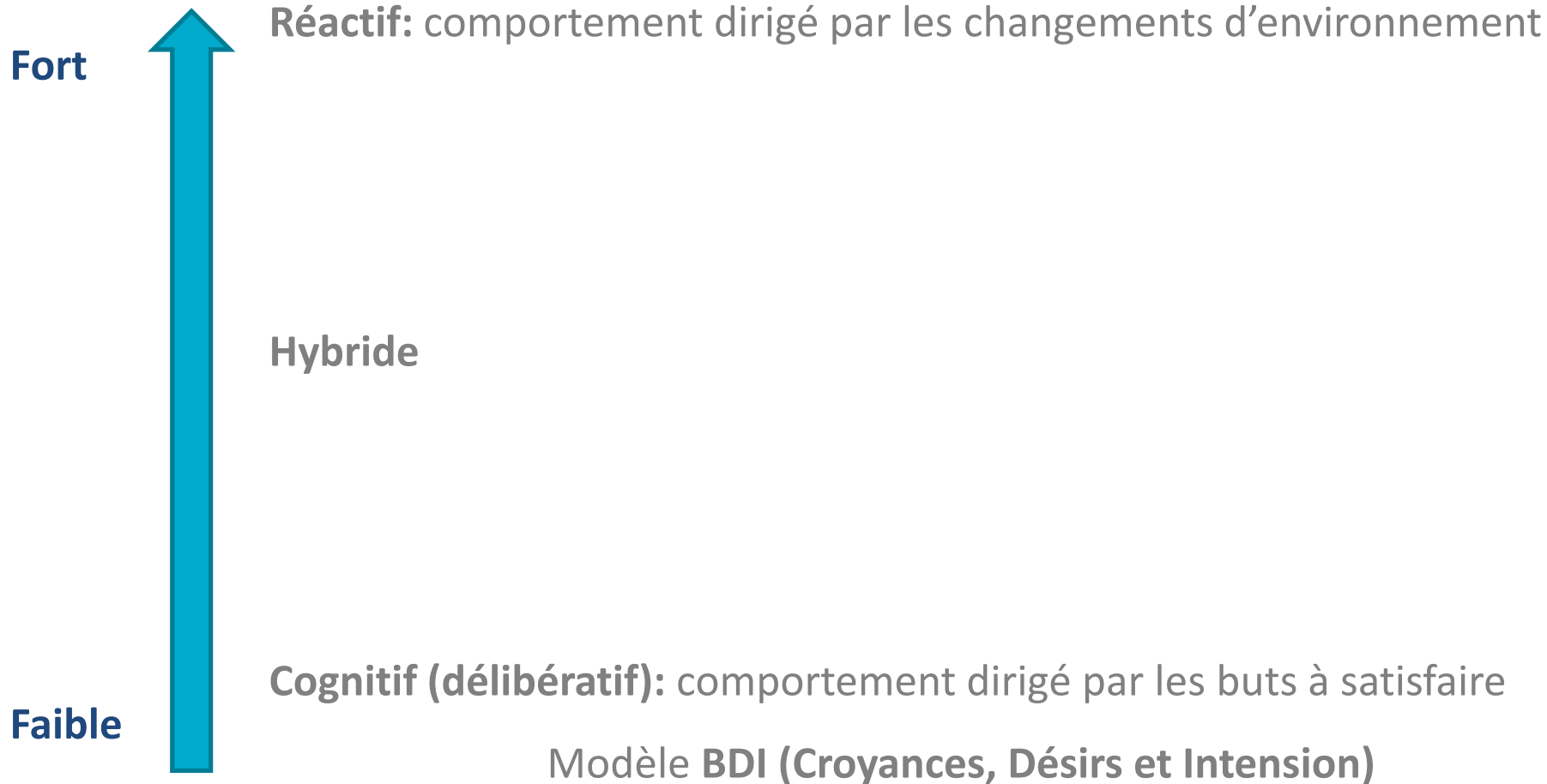
Agent cognitif

Agent muni d'une **base de connaissances** nécessaire à la **réalisation de sa tâche** ainsi qu'à la **gestion des interactions** avec les autres agents et **avec son environnement**

Agent hybride

Compromis entre un agent purement réactif et un agent purement cognitif, visant à **concilier les capacités intéressantes des deux types d'agents précédents**.

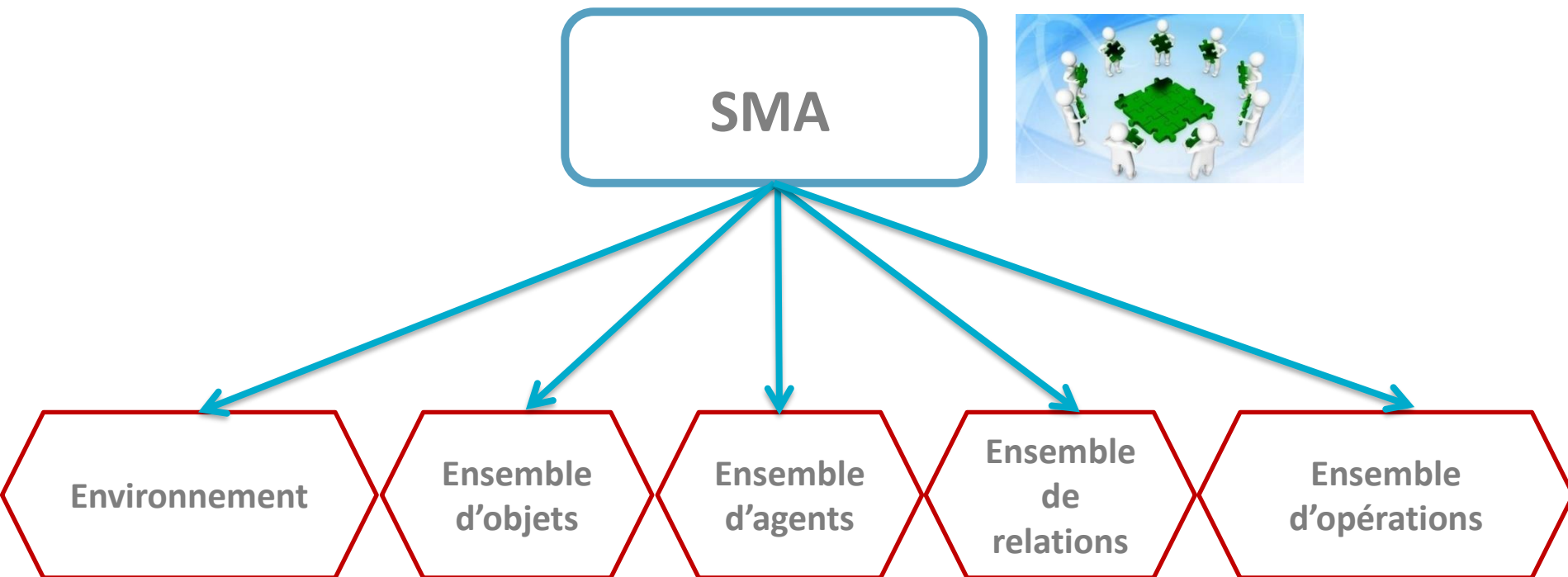
Typologie des agents



Couplage à l'environnement

Systemes Multi-agents

Ensemble d'agents agissant et interagissant dans un environnement commun



Interactions entre agents

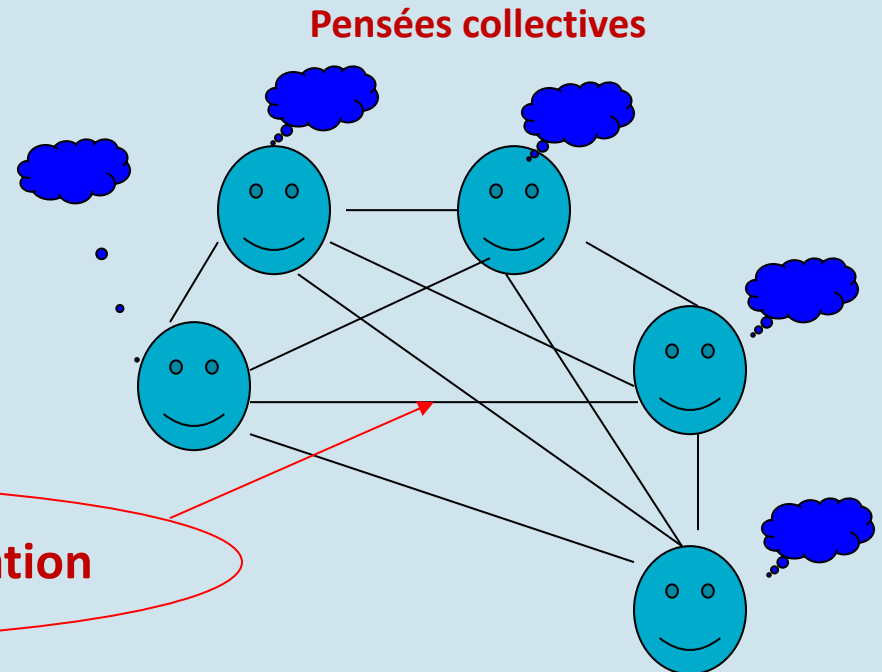
- Un agent doit être capable de communiquer avec les autres agents
- Les agents doivent avoir des capacités à manipuler un langage commun
- Deux types de communication :
 - **Communication indirecte** : partage d'informations via l'environnement
 - **Communication directe** : envoi de messages

Intelligence collective

- On est passé de la métaphore du penseur isolé



- A celle de communauté de penseurs

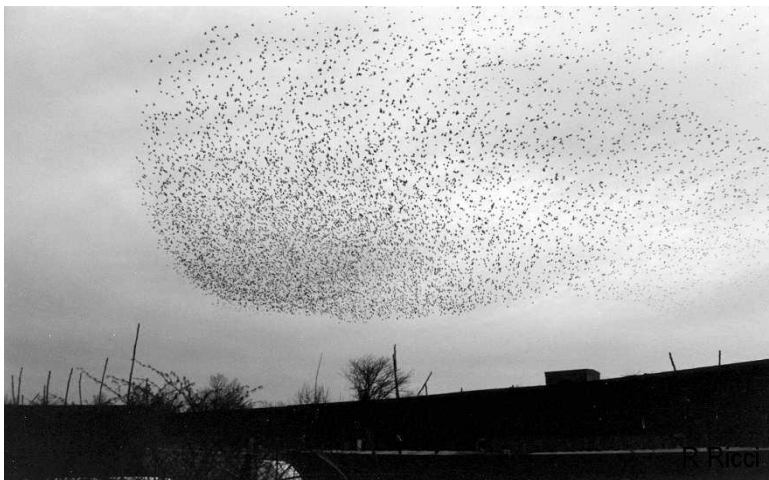


→ **D'abord : interaction = coopération**

Intelligence collective

Connaissez-vous des cas d'intelligence collective naturels ?

Groupements naturels



Introduction

I. Agents et environnements

II. Concept de rationalité

III. Environnements

IV. Systèmes multi-agents

V. Applications & plateformes de développement

Conclusion

Applications : évacuation du public d'un stade



<https://www.youtube.com/watch?v=c6unBZoY9Ag>

Plateformes de développement des SMA

JADE : Java Agent DEvelopment Framework



Madkit : The Multiagent Development Kit



AgentBuilder : Environnement de développement complet



JADEX : modèle d'agents BDI s'appuyant sur JADE

PADE, SPADE



Introduction

I. Agents et environnements

II. Concept de rationalité

III. Environnements

IV. Systèmes multi-agents

V. Applications & plateformes de développement

Conclusion

Conclusion

- Un **agent** perçoit et agit dans son environnement
- La **fonction d'agent** : décrire le comportement de l'agent
- Le **programme d'agent** : implémente la fonction d'agent
- **Mesure performance** évalue comportement d'agent dans son environnement
- **Agent rationnel** : maximiser la mesure de performance
- Environnement décrit par la description **PEAS**
- **SMA** = Agents + Environnement + Interactions + Organisations (AEIO)
- Les agents peuvent apprendre aussi : **Agents d'apprentissage**

→ prochains chapitres

Un peu de pratique pour terminer

A rendre via Moodle avant le 28/02/2025 : travail individuel

- a. Visionner cette [vidéo](#) montrant la création de système multi-agents utilisant les LLMs
- b. Résumer votre analyse du projet en **01 page (minimum)** décrivant les points suivants:
 - Décrire l'objectif de ce système multi-agents de création de newsletters ;
 - Quel est le rôle de modèles LLMs dans ces systèmes multi-agents ;
 - Quel lien voyez-vous entre les systèmes multi-agents et méthodes d'apprentissage.
 - **Facultatif** : tester la solution avec [flowiseai](#) (avec le version d'essai de 14 jours).

Travail pratique bas (facultatif)

- Pour prendre en main le développement d'agents (**fortement conseillé**):

→ **TP 0 : Manipulation des agents avec la plateforme SPADE de python**

[Enoncé partagé via Moodle](#)

Références

- [1] S. Cammarata, D. McArthur, and R. Steeb. “**Strategies of cooperation in distributed problem solving**”. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, Karlsruhe, Germany, 1983
- [2] Jacques Ferber. “**Les systèmes multi-agents, vers une intelligence collective**”. InterEditions, 1995
- [3] Jacques Ferber, “**Multi-Agent System: An Introduction to Distributed Artificial Intelligence**”, Harlow: Addison Wesley Longman, 1999
- [4] Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood, “**Developing Multi-Agent Systems with JADE**”, John Wiley & Sons Ltd, 2007
- [5] R. Olfati-Saber and J. A. Fax and R. M. Murray, “**Consensus and Cooperation in Networked Multi-Agent Systems**”, *Proceedings of the IEEE*, vol. 95 (1), pp. 215-233, 2007
- [6] Pathfinder, “**Agent Based Evacuation Simulation, Advanced movement simulation combined with high-quality 3-D animated results, gives you reliable answers quickly**”, 2017
- Etc.

MERCI