

Course 2: Basic principles: Multi-Agent Systems (MAS)

I. Introduction

An agent perceives the environment using its sensors and it acts on its environment using its effectors. So, an agent can always be seen as a function linking its perceptions to its actions.

We obtain a typology of agents if we answer the question: why does an agent act?

Answer: how to implement the actions efficiently according to the current situation of the environment or other agents. Multi-agent systems are generally classified into two main families:

- cognitive systems
- and reactive systems. We try to understand:
- Why agents are an important new approach to designing and implementing some software applications
- What are agents and what are the links with other software paradigms, in particular, expert systems and object-oriented programming?

II. Agents

II.1. Definition of an agent

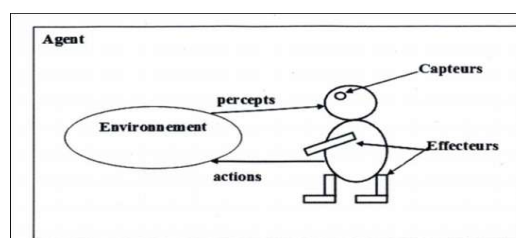
In the literature, there are a multitude of definitions of agents. They differ depending on the type of application for which the agent is designed.

According to J.Ferber, An Agent can be defined as:

- an autonomous entity (real or abstract)
- able to act on itself and its environment
- having a partial representation of this environment
- able to communicate with other agents (in a multi-agent universe)
- whose behavior is the consequence of his observations, knowledge and interactions with other agents.

According to Russell and Norvig, an Agent can be defined as any entity:

- having the ability to perceive its environment through its "sensors" (sensory inputs)
- and to act on such an environment via its "actuators".



- Human agent “Human agents”: (eyes as sensors, hands as actuators)
- Robotic agents: (cameras as sensors, wheels as actuators)
- Software agent “Software agents”: (a graphical user interface as sensor and as actuator)

II.2. Intelligent Agent

According to Sycara and Wooldridge: An Intelligent Agent:

- is a computer system located in an environment
- and it acts autonomously and flexibly
- to achieve the purposes for which it was designed,

Flexibility?

- **Reactive:** ability to perceive the environment and respond in time (develop a response within the required time) to changes that may affect the environment;
- **Pro-active:** goal-oriented behavior by taking initiatives at the “right” time;
- **Social (Social Aptitude):** ability to interact with other agents in a cooperative or competitive way to achieve its objectives.

II.3. Multidimensional characteristics

Some properties emerge from these definitions:

- **Nature:** physical or virtual agents,
- **Situated:** the agent is able to act on its environment from the sensory inputs it receives from this same environment (process control systems, embedded systems, etc.).
- **Autonomy:** the agent is able to act without the intervention of a third party (human or agent) and controls its own actions as well as its internal state (resources);
- **Perception:** of the environment by the agent.
- **Communication:** the agent has the ability to communicate with other agents as well as with human users.
- **Reasoning:** the agent can be linked to more or less complex reasoning mechanisms (equipped with inferential rules).
- **The environment:** this is the space in which the agent will act; this can be reduced to the network constituted by all the agents.
- **The representational capacity:** the agent can have a very local vision of its environment, but it can also have a broader representation of this environment and in particular of the agents which surround it (acquaintances).
- **Intentionality:** an intentional agent is an agent guided by its goals. An intention therefore expresses the will of an agent to achieve a goal or perform an action.

- **Rationality:** rational agents have criteria for evaluating their actions, and select according to these criteria the best actions to achieve the goal.
- **Adaptability:** an adaptable agent is an agent capable of controlling its aptitudes (communicational, behavioral, etc.) according to the environment
- **Commitment:** The notion of commitment is one of the essential qualities of cooperative agents. A cooperative agent plans its actions by coordinating and negotiating with other agents. By constructing a plan to achieve a goal, the agent gives himself the means to achieve it and therefore undertakes to perform the actions that satisfy this goal; the agent believes that he has elaborated, which leads him to act accordingly.
- **Learning:** an agent can memorize its experiences and adapt its behavior accordingly.
- **Mobility:** an agent can move from one machine to another and possibly duplicate itself.

- thermostat avec détecteur de température	# situé
	# autonome
- calendrier électronique	# proactif
- courrier électronique trié par expéditeurs	# capable de répondre à temps
- courrier électronique trié par importance	# social
- le système de control aérien de l'aéroport X	# buts
	# coordination
	# communication
	# organisation adaptation

II.3.1. Agent properties (autonomy)

The objective of autonomy is robustness. Agents have control over their own internal states and behaviors. Behavioral autonomy is:

- Make decisions “by yourself”
- Have the power to say no!
- Do not depend on others (at least for decisions) But also relates to resources:
- To act, the agent needs a certain number of resources: energy, CPU, quantity of memory, access to certain sources of information,
- while being able to manage these resources himself,

II.4. Comparison with other technologies (objects)

The agent paradigm is distinguished from conventional systems such as object-oriented systems and expert systems.

Object: informatic entities that:

- Encapsulates some state (property)
- Communicate by sending a message

- capable of performing actions or methods on this state
- **Difference:** an agent, like an object, encapsulates a state and a behavior, BUT ? ? ?

Object vs. Agent: Degree of Autonomy

- An object has autonomy over its own internal state: it has control over this state, Example: In JAVA, you can declare private variables (or methods): accessible only from inside the object. They can also be declared as Public: accessible from anywhere,
- But an object has no control over its own behavior: its methods are executed without restriction when invoked.
- The agent has the power to negotiate or refuse requests for services (request for execution of actions)
- So: In the object-oriented case, the decision depends on the object that invokes the method. in the agent-oriented case, the decision depends on the agent receiving the request

Object vs Agent: Flexibility: flexibility is absent for objects.

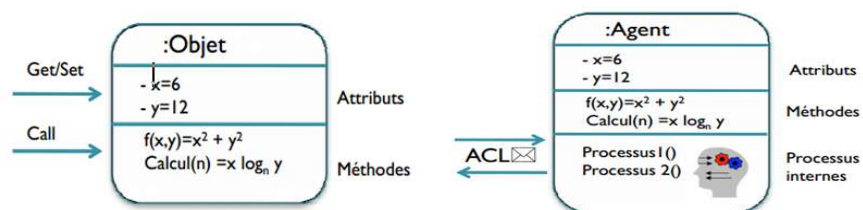
Object vs Agent: Control Flow:

- an agent is assumed to have its own control path.
- exercises this control in different ways (reactive, goal-directed, social)
- has its own execution thread (thread), so an MAS has several control flows (multi-thread)
- object system has only one control flow

An object is reactive: If nobody asks for the value of an attribute or activates a method of the object, then nothing happens, so An object is a passive (or reactive) entity.

An agent is Proactive: An agent has internal processes that work even in the absence of external solicitations. An agent can therefore act even if no one asks him for anything.

The concept of active objects makes it possible to make agents but not to make them intelligent (flexible and autonomous). So object-oriented languages can help build agents...



II.5. Comparison with other technologies (expert systems)

Expert systems: is software capable of answering questions, by performing reasoning based on known facts and rules. It can be used in particular as a decision-making tool. An expert system consists of three parts:

1. A base of facts;
2. A rule base;
3. An inference engine.

Agent vs expert systems:

- does not act on any environment, provides guidance
- reasons, but does not act in perception of..., nor in action on...
- does not obtain information through sensors, but through a user acting as an intermediary.
- not autonomous: require interventions and instructions.
- Not pro-active, adaptive, distributed.
- ES is not supposed to cooperate with other agents (except some real-time SE like ARCHON)
- The evolution of ES towards multi-expert systems was at the origin of MAS.

III. Multi-agent systems (MAS)

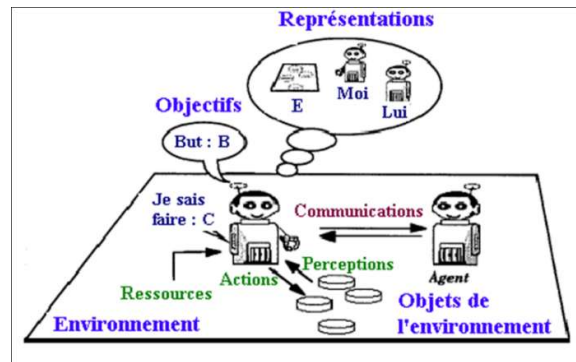
A multi-agent system is a system that consists of several organized agents, which interact or communicate with each other in a common environment to achieve a global objective. In SMAs:

- the agents are (among other things) autonomous, possibly pre-existing and generally heterogeneous.
- To interact, they must have the ability to: cooperate, compete or coexist, coordinate, negotiate among themselves.

III.1. The elements of an SMA

Ferber offers a general description of the structure of a Multi-Agent system:

1. **an environment:** ie a space generally having a metric.
2. **a set of located objects,** ie it is possible to associate each object, at a given time, with a position in the environment.
3. **a set of agents** representing active entities of the system.
4. **a set of relations** which unite objects (therefore agents) with each other.
5. **a set of operations** allowing agents to perceive, produce, consume, transform, manipulate objects.
6. **operators** responsible for representing the application of these operations and the relation of the world to this attempt of modification



III.2. Characteristics of an SMA

An SMA is usually characterized by:

- Each agent has limited information or problem-solving capabilities (abilities),
- Thus, each agent has a partial point of view,
- The calculation is asynchronous.
- It is made up of one or more organizations that structure the rules of cohabitation and collective work between agents.
- In the same system, an agent can belong to several organizations.
- Total distribution of knowledge (data is decentralized) and control (no global control of the multi-agent system).

III.3. Advantages and objectives

MAS are ideal systems for representing problems with multiple solving methods. The S.M.A approach is justified by the properties:

- Modularity and extensibility.
- Speed, with parallelism.
- Reliability and fault tolerance due to redundancy and distribution of control with shared responsibilities.
- Symbolic processing at the knowledge level.
- Reusability and portability.
- The intervention of sophisticated interaction schemes (cooperation, coordination, negotiation).

The two major research objectives in the field of S.M.A are:

- theoretical and experimental analysis of the mechanisms.
- The resolution of distributed programs.

To solve two kinds of problem:

1. Model, explain and simulate natural phenomena.
2. The realization of complex computer systems.

III.4. Challenges and Constraints

Although MASs offer many potential benefits, they also face many challenges:

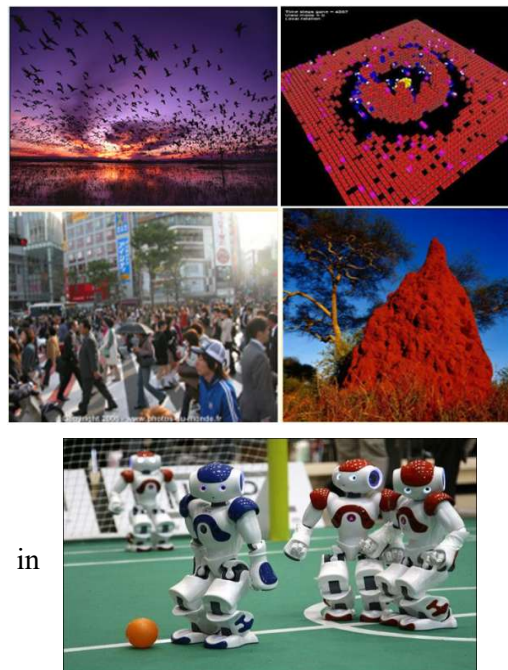
- How to formulate, describe, break down, and allocate the problems and synthesize the results?
- How to enable agents to communicate and interact? What and when to communicate?
- How to ensure that the agents act in a coherent way: by taking their decisions or actions, by managing the nonlocal effects of their local decisions or by avoiding harmful interactions?
- How to allow agents to represent and reason about the actions, plans and knowledge of other agents in order to coordinate with them?
- How to reason about the state of their coordinated processes (such as initialization or termination)?
- How to recognize and reconcile disparate points of view and conflicting intentions in a set of agents trying to coordinate their actions?
- How to find the best compromise between local processing at the level of a single agent and distributed processing between several agents (distributed processing which induces communication)? More generally, how to manage the distribution of limited resources?
- How to avoid or reduce harmful behavior of the global system, such as chaotic or oscillatory behavior?
- How to design technological platforms and development methodologies for MAS?

III.5. Application fields

3 main families of problems

III.5.1. Simulation:

- Studies of complex phenomena of the real world: ethology (social animals), sociology, economy, environment,
- Represent and simulate social or natural systems involving a large number of individuals, simulation of complex systems:
- Ecosystem with fishermen + schools of fish + pollutants
- Rats and the pool
- Insect societies: nest building and foraging in ants, social spiders, termites, bees,...etc.



- Formation of a traffic jam, opinion dynamics.
- Collective robotics: Several robots cooperating to accomplish a mission

Understanding of interactions between humans: behavioral simulations - explanation of the impact of individual behavior on the global level:

- Customer study
- Integration of mentally ill people
- Model virtual environments: (massively) multiplayer video games (character intelligence), cinema (Massive software) (Used for the creation of film special effects (crowd effects))



Training on environments for collective activities

III.5.2. Collective problem solving:

- distributed process control without centralization:
- supervision of a production workshop: several machines, several parts, several machining operations, . . . organization ?
- Automatic reorganization in the event of a breakdown.
- supervision (Monitoring) of telecommunications networks (detection, intervention, repair): monitoring agents, fault diagnosis agents, transport network maintenance operator agent (energy, . . .). . . etc

III.5.3. Integration and make software interoperate with human beings and mechanical systems:

- Commerce: B2C, B2B applications, purchasing processes with sophisticated negotiations (automated auctions, bilateral negotiation, of contracts: buyer or seller agent who represents the buyer, seller with different types of behavior)
- Research and filtering of information on the Web: aggregation of services (ex: travel organization: train ticket, hotel room, museum ticket, etc.),
- Diffuse computing or ambient intelligence, pervasive computing: move towards an open computing world, sensors in the environment (smart clothes, ad hoc networks, etc.)

IV. Environment

Rational agent:

For each possible percept sequence, a rational agent should select an action that should maximize its performance measure, taking into account the knowledge of the agent. The

rationality of the agent can then be evaluated by a performance measure taking into account the following aspects:

- The performance measure to define the success of the agent.
- agent's prior knowledge of the environment.
- The actions that the agent can perform.
- The agent's percept sequence in time.

Example: The vacuum cleaner world

Perceived: location and content, e.g., [A,Dirty]

Actions: Left, Right, Suck, NoOp

Performance Metric: An Objective Criterion of Success for Agent Behavior

- volume of dirt picked up,
- time taken,
- electricity consumed,
- noise produced, etc.

IV.1. Perception

The agents can observe different things from the environment surrounded by their sensors

- the observed data differs spatially (appearing in different places), temporally (arriving at different times), or semantically (requiring different interpretations),
- makes the world partially observable to each agent

Another problem is sensor fusion:

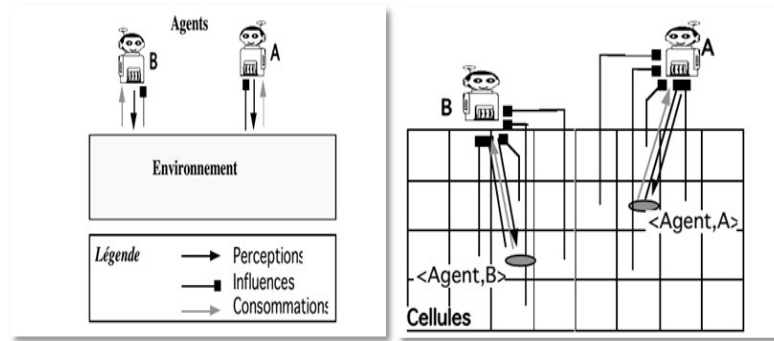
- how agents can optimally combine their perceptions in order to increase their collective knowledge about the current state
- **In the case of simulation:** a sensor is an interface between the environment and the agent.
- **In the case of problem solving:** perception is a mechanism for selecting a point of view on the problem.
- **In the case of integration and collaborative systems:** perception is an interpretation of external information.

IV.2. Environment

The environment is seen as a problem in which agents act as solvers. This constitutes an essential part (element) of situated multi-agent systems. The environment can be seen as being in a state “e” among a set of states $E = \{e_1, \dots, e_n\}$,

- The environment can change its state either spontaneously or as a result of the agent's actions.

- It is possible to represent it in two ways:
 - considering it as a monolithic block: centralized environment
 - by modeling it as a set of cells brought together in a network (cellular automaton): distributed environment.



Environment of the MAS: common space to the agents of the system.

- This environment is equipped with a set of located objects (determining the position of an object), passive (these objects can be perceived, destroyed, modified by agents) or active.
- plays an important role in the behavior of an agent:
 - memory in which different traces are left,
 - source of feedback to the agent,

Agent environment: environment of the MAS and the other agents belonging to the system.

IV.3. Properties

Russell and Norvig propose some properties of an environment:

IV.3.1. Accessible vs Inaccessible: Completely Observable, Partially Observable

Can the agent obtain complete and precise information about the environment?

- **Completely observed environment**
 - if the sensors of an agent give it access to the entirety of the environment's state at any moment.
 - if the sensors detect all the aspects that are relevant for the choice of action.
 - Relevance: The quality of an agent's decision depends on the quality of the available information.
 - Poor information \Rightarrow poor decision by the agent.
 - If the agent has no sensors at all, then the environment is unobservable.

Examples

- The game of chess is completely observable: one can see the position of all the pieces.

- The game of poker is partially observable: one does not know the cards in the opponent's hand.
- **Accessible task environment** is one in which the agent can obtain complete, precise, and up-to-date information about the state of the environment.
- Most moderately complex environments (for example, the Internet) are inaccessible.
- If the environment is not completely accessible, the agent will need internal states.

IV.3.2. Deterministic / Non Deterministic (stochastic)

Is each action guaranteed to produce a unique effect?

Est-ce que chaque action est garantie de produire un effet unique ?

- Si l'état suivant de l'environnement est complètement déterminé par l'état courant et par l'action qu'exécute le ou les agents, on dit que l'environnement est déterministe
 - sinon, il est non déterministe « stochastique »
- Un environnement déterministe est un environnement dans lequel chaque action a un seul effet garanti

Exemples

Le jeu des échecs est déterministe : déplacer une pièce donne toujours le même résultat

IV.3.3. Regular / Non-Regular: (episodic/ sequential):

Does the next action of an agent depend only on the current state of the environment (episodic), or on an evaluation of past states of the environment (sequential)?

- In an episodic task environment, the sequence (perceptions/actions) of the agent consisting of atomic episodes
- Each episode consists of observing the environment (receives a percept) and performing a single action
- this action has no influence on the environment in the next episode
- the next episode does not depend on the actions performed during the previous episodes

Examples:

- Character recognition is episodic: the system's prediction does not influence the next character to be recognized
- Poker Game: Deciding whether I bet or not has an impact on the next state of the game

IV.3.4. Static / Dynamic

- **dynamic environment:** the environment can change while an agent is thinking; otherwise, it is static.
- **Environment is semi-dynamic:** with the passage of time; the environment itself does not change but the performance score will change.

- **Static environments:** the agent does not need to observe the world while it chooses an action, nor the time that passes.

Example:

- Crossword puzzles are static.
- When using a stopwatch, the game of chess is semi-dynamic.
- Driving a taxi is clearly dynamic: other vehicles and the taxi itself continue to move forward,

IV.3.5. Discrete / Continuous:

- If there is a fixed and finite number of perceptions (sensory data) and actions, the environment is discrete.

Example: the game of chess is a discrete environment

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Chess with a clock	Yes	Yes	No	Semi	Yes
Chess without a clock	Yes	Yes	No	Yes	Yes
Poker	No	No	No	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi driving	No	No	No	No	No
Medical diagnosis system	No	No	No	No	No
Image-analysis system	Yes	Yes	Yes	Semi	No
Part-picking robot	No	No	Yes	No	No
Refinery controller	No	No	No	No	No
Interactive English tutor	No	No	No	No	Yes

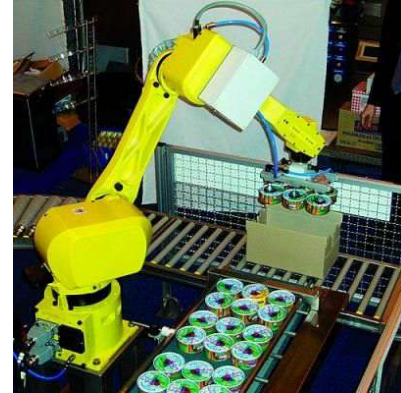
IV.4. PEAS description:

The first step when designing a rational agent is to specify the task environment by the following four elements (PEAS):

- Performance: Performance measurement
- Environment
- Actuators
- Sensors

Example: Agent: robot for storing parts

- **Performance measure:** Percentage of parts put in correct boxes
- **Environment:** Parts conveyor, boxes
- **Actuators:** mechanical arm
- **Sensors:** Camera, angle sensors



Type d'agent	Mesure de performance	Environnement	Effecteurs	Capteurs
Système de diagnostic médical	Rétablissement des patients, minimisation des coûts	Patient, hôpital, personnel	Affichage de questions, tests, diagnostics, traitements, orientation	Entrée au clavier de symptômes, recherches, réponses du patient
Système d'analyse d'images satellites	Catégorisation correcte des images	Liaison descendante depuis un satellite en orbite	Affichage de la catégorisation de la scène	Tableaux de pixels colorés
Robot contrôleur de pièces	Pourcentage de pièces dans les bonnes corbeilles	Tapis roulant avec pièces, corbeilles	Bras et main articulés	Caméra, capteurs articulés
Contrôleur de raffinerie	Pureté, production, sécurité	Raffinerie, opérateurs	Valves, pompes, réchauffeurs, affichages	Capteurs de température, de pression et chimiques
Répétiteur d'anglais interactif	Notes des étudiants aux contrôles	Ensemble d'étudiants, organisme faisant passer les tests	Affichages d'exercices, de suggestions, de corrections	Entrées au clavier

Figure 2.5 : Exemples de types d'agents et leurs descriptions PEAS.