REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITE IBN KHALDOUN - TIARET**

# MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité: *Génie Informatique*

Par :

**SEBTI Sarra**
**MEHIZEL Sara Nour Elyakin**

Sur le thème

# Les Systèmes Multi-Robot coopératifs : Développement d'un lexique de communication

Soutenu publiquement le 08 / 10 / 2020 à Tiaret devant le jury composé de :

| | | | |
|---|---|---|---|
| Mr MERATI Medjeded | MCB | Université de Tiaret | Président |
| Mr MOKHTARI Ahmed | MAA | Université de Tiaret | Encadreur |
| Mr BERBER El-Mehdi | MAA | Université de Tiaret | Examinateur |

2019-2020

# Acknowledgement

First and foremost, praises and thanks to Allah, the Almighty and Merciful who gave us strengthand patience to complete the thesis successfully.

We address our most sincere gratitude to our promoter, framer and thesis advisor Mr. MOKHTARI Ahmed, for having proposed this subject to us, for his availability, support along this work and for the help and knowledge he has provided us.

We also want to take this opportunity to thank they jury's members : Mr. MERATI Medjeded and Mr. BERBEREl-Mehdi for having honored us by accepting to judge and to evaluate this work.

Finally, we are grateful to all those who contributed in this work by their advice or encouragement.

# Dedications

# SEBTI

For that women who passed the last night praying and if you went to see her now she stills praying the women who was always there for me, mom I'll make you proud of me as I promised.

Dad

Although you never get emotional I saw how proud you are in your eyes ,I'll work so hard to deserve that look of pride for the rest of my life. your support did always meant the world for me if I got the chance to choose a dad for billion of times I'll choose you each and every time.

For all my brothers and sisters

Nadjiba .Khedidja,Messouda,Souhila,Ouanissa,Abdallah,Mohamed,Ibrahim and the little Houcine.

For the soul of my dearest friend Sara Nour El Yakin Mehizel s' father .

For Wassim who nothing of this work could be done without his help.

For all my friends for their tremendous support Sarah ,Imen,Michelle,Sarra,Hanane,Randa,Rania ,Khedidja,Fathia,Ikram,Nadjla ,Houria,Djamel,Ilyes ,Kadda….

Thanks for all who has believed in me and supported me all the time long.

# Dedications

# MEHIZEL

My part of this work is dedicated to my beloved deceased father, someone who meant so much to all those he knew, who left a tremendous void and precious memories too. Dady I have always wanted to make you proud and words can't convey how much i wish that you were here with me today to make you proud once more. May Allah have mercy on your soul and may it rest in Allah's eternal peace.

To my mother, the woman made of steel, our shelter and protector. You have been the most perfect mother that can anyone wish to have. I may have failed you once or twice but i hope this work would make you proud of me again. I love you to death.

To my sister who was nothing but supportive, I hope i made a good role model to you, i wish you the best in your life, be strong like i always taught you. I love you.

To my brother whose words are few but meaningful to me, thank you for your support, I wish you success in the Bac exam. I love you.

To my family, aunt Fatiha « Yaya », aunt Djamila, uncle Noureddine, uncle Kadda and my cousins : Amina, Imen, Sabah, Khouloud, Rachida, Fatiha and Mohamed.

To my partner and greatest friend Sarra, you were nothing but understanding and supportive, i leaned on you and you didn't disappoint, i swear i only carried on working for you. Thank you.

To my beloved friends : Dhiae Eddine, Fatiha, Z'hor, Oussama and Fatiha

My most sincere and greatest gratitude to all those who supported and encouraged me.

# Table of contents

# ABSTARCT

Social learning, which enables individuals to learn fromeach other, is a powerful mechanism in social animals, including humans, that doesn’t differ much for artificial agents as they represent the replacements of humans. It could be difficult for these agents to communicate in an unknown environment for they don’t perceive nor categorize the objects in the same way.

The concept of Language-Naming games is a mechanism to allow agents to self-organize in order to develop a shared and common lexicon, so they have to coordinate and cooperate to name the encountered objects corehently and non-ambiguously.

To successfully communicate,agents must have access to the shared memory that contains perceptions and their assigned meanings.

**Keywrods :**

Artificial agents, communication, cooperation, percision, categorization, language games, naming games, common lexicon, meanings, perceptions ….

# List of figures:

# List of tables:

# List of acronyms:

**MAS**: Multi-agents’ system
**MRS**: Multi-robots’ system

**LG**: Language game

**MNG**: Minimal Naming game

**UML**: Unified Modeling Language

**HIM**: Human Interface Machine

# GENERAL INTRODUCTION

# GENERAL INTRODUCTION:

"Technology is a way of revealing. If we give heed to this, then another whole realmfor the essence of technology will open itself up to us. It is the realm of revealing,i.e., of the truth." – Martin Heidegger **[1]**

The basics of the world we live in are the material implementations of philosophical and mathematical abstractions, while technology in robotics specifically - robots, and especially mobile robots andmobile robot systems, implies our imagination and sense of motion.

The idea of robots helping humans is a constant source of inspiration for researchers all over the world, soreaching for the human frontier is a constant challenge in different domains for robotics; Interacting, exploring, planning, foraging[1], navigation, cooperation and communication i.e. robots cooperate autonomously[2] to fulfill a specific task in an uncertain environment.

## - *Context and motivation:*

Direct communication in multi-robots' systems is used in simple tasks such as signaling robots' preferences and states. This was inspired by the emergence of meaning found in natural languages because if the communications skills were more complex the design of robot swarm would be more autonomic and adaptive. The context of our work revolves around developing a common lexicon within a swarm of robots in order for them to communicate and get tasks done.

## - *Problematic (leading argument):*

Inside a population of autonomic robots:

It is difficult to develop a common lexicon to specify different objects in an uncertain environment.

How can the words of this lexicon refer to shared meanings when the different sswarm robots cannot perceive nor categorize the world in the same way? (Communication).

## - *Objective:*

The objective of this work, is to offer a way to understand how multi-robots' systems (specifically swarm robotics) cooperate in language development/evolution and communication. It is divided into two sections :

1) Emerge/develop a conventional lexicon to specify objects in the robots' environment. (Language development).
2) Each word (description) must relate to a single and a specific meaning (perception) so that different concerned swarms could categorize their specified objects and understand each other. (Communication)

## - *Thesis' structure (Organization):*

Our thesis is organized into four chapters:

### *Chapter 01: Multi-agents' systems:*

This chapter is devoted to describing general concepts on multi-agent systems.

***Chapter 02:*** ***Multi-robots' systems and Swarm robotics ( Language development and communication):*** In this chapter we discussed the key concepts on multi-robots’ systems, swarm robotics, language development and the use of the later in an uncertain environment (communication).

***Chapter 03:* Conception:** Through this chapter, we have given details about how the language development works in giving pseudo codes thus we use UML tools to clarify how the communication happens.

***Chapter 04:* Implementation:**

This chapter is devoted to show the technologies and tools used to realize our system and illustrations (screenshots) to show how the system works.

This figure illustrates the visualization of our work:

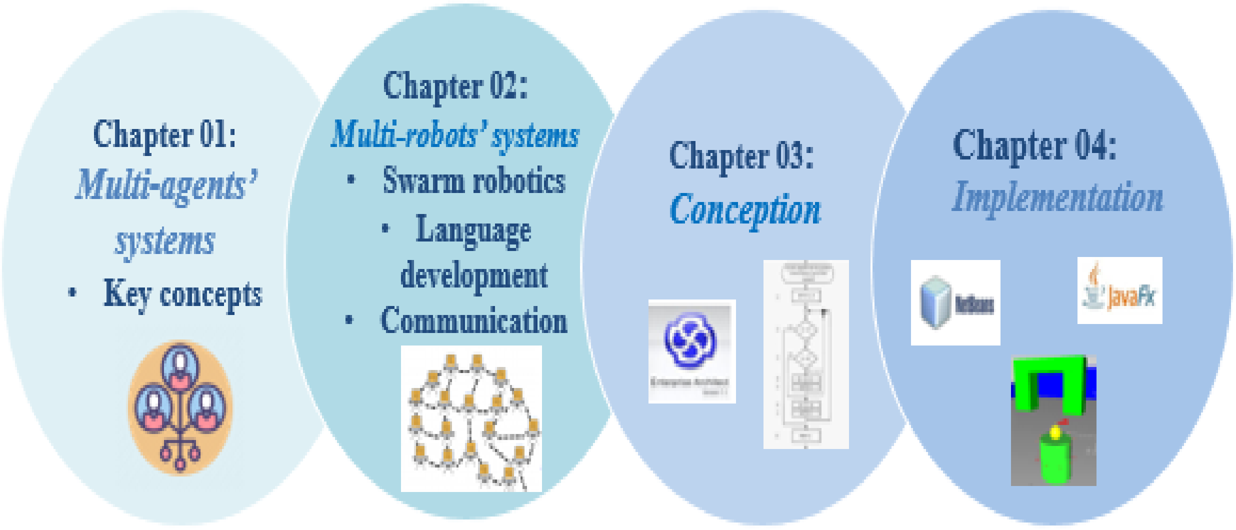


Figure 1: Organization of the thesis

# CHAPTER 01 : MULTI-AGENTS' SYSTEMS

# I. CHAPTER 01: MULTI-AGENTS' SYSTEMS

## I.1.Introduction:

Where did the "multi system" term come from ?

To respond to that question we mention five basic trends in the history of computing which are:

- **Ubiquity:** Computer processing power is everywhere.
- **Interconnection:** the idea that these processes can communicate with one another.
- **Intelligence:** they are capable of solving more and more complex tasks.
- **Delegation:** the idea of handing over control to these processes.
- **Human orientation:** The way we interact with computers resemble the way we interact with people.

Besides after the huge programming progress through time, all the way from machine code and assembly language to objects, this lead us to agents where the object's concept came from the reality of objects we use in daily life and since it's used by people "agents" in real life, the scientists thought of creating agent software to manipulate programming objects.

This chapter is devoted to define what is an agent and its role in a multi-agent system, its architecture, its properties thus the different interactions and communication .

## I.2.Agents and Multi agents:

### I.2.1.Agent (definition):

It is a computer system which is acting to carry some delegated tasks in semi-intelligent way. We cannot attribute a single definition to the notion of "agent", because this term is used in many different applications by communities from diverse backgrounds. The latter has, since its appearance, been enriched and refined, in particular by **Wooldridge** and **Jennings 1995[2], Ferber 1995** and others but they did not agree on a single description.

We call agent a physical or virtual entity:

- *Ferber, 95 [3] :*

An agent is a real or virtual entity, evolving in an environment, capable of perceiving and acting on it, who can communicate with other agents, who exhibits an autonomous behavior which can be seen as the consequence of its knowledge and interactions with other agents and the goals they pursue.

- *Demazeau, 95 [4] :*

An agent is a real or virtual entity, whose behavior is autonomous, evolving in an environment, is capable of perceiving, acting and of interacting with other agents.

- *Wooldridge, 98 :*

An agent is a computer system capable of acting autonomously and flexibly in an environment. Flexibility means responsiveness, pro-activity, adaptability and social skills.

### I.2.3.Multi-agent systems (definition):

A multi-agent system is one that consists of a number of agents which interact with one another, in the most general case agents will be acting on the behalf of users with different goals and motivations.

To successfully interact and achieve their delegated goals, they will require the ability to cooperate, coordinate and negotiate with each other much as people do

## I.2.4.Two Key Problems:

### I.2.4.1. *Agent design:*

It is how to build agents that take actions autonomously in order to achieve the delegated task.

### I.2.4.2. Society design:

It is how to build agents that cooperate and negotiate to achieve a specific task even though the goal is not similar for all the agents.

## I.2.5. Intelligent Agent:

### I.2.5.1. Agent and environment:

There's a quiet dynamic relation between the agent and its environment where every agent precises its environment through sensors that allows it to make a decision, so it acts and gives feed backs in return about the new status of the environment after the action and so on.

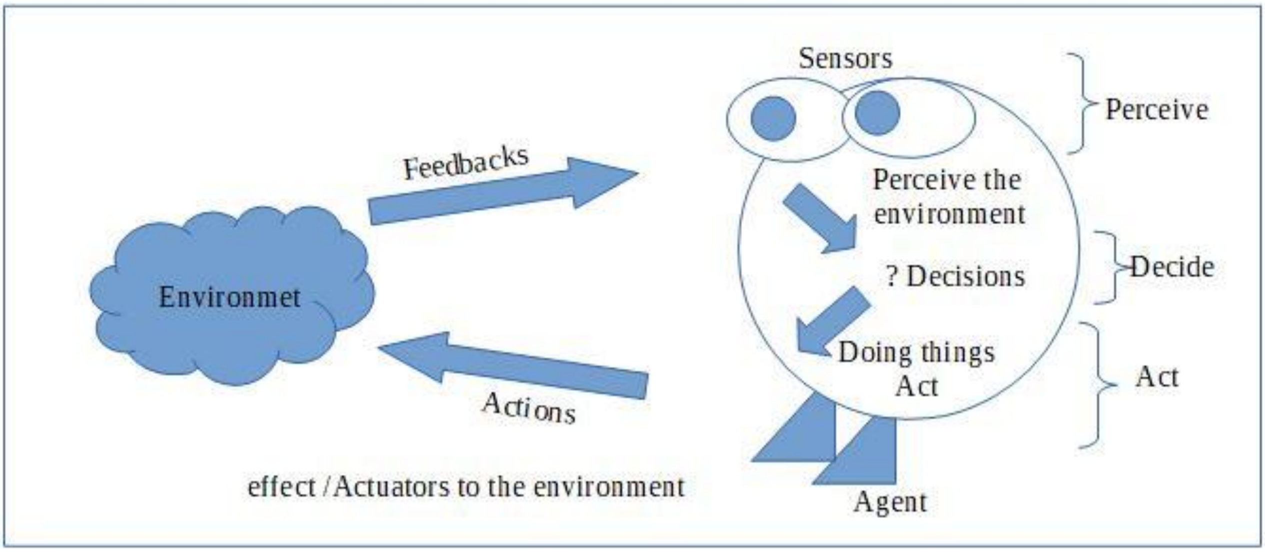


Figure 2: Agent and environment

We think of an agent as a being in a close complicated continual interaction with its environment:

- **Sense:** look into the environment and get information.
- **Decide:** a basis of information about what they decide to do next.
- **Act:** perform an action that changes the environment.
- **Sense:** see what the effect of the action - on the environment is.
- **Decide**: deciding according to the sense they've got.

### I.2.5.2. Properties of intelligent autonomous agents:

Intelligent agents exhibit three types of behavior:

#### *I.2.5.2.1. Reactivity:*

It is realizing when the world is changing so it makes the plan unachievable and responds to "reorganizing" activities in order for those consequences to be useful (to find other plans that guide to the same goal). This is very important because most environments where we put agents, are dynamic and the changes that happen are beyond the agent's goal-directed behavior and control.

#### *I.2.5.2.2. Pro-activity:*

Exhibiting goal-directed behavior is being capable of working towards a goal (delegating goal) and then deciding how best to try to achieve that goal (systematically working to achieve that goal).

#### *I.2.5.2.3.* Social ability:

Is to communicate and to collaborate as humans do (to collaborate if they have the same goal and negotiate in case of conflating).

### I.2.5.3. Agent and object:

Are agents just objects by another name?

### *I.2.5.3.1. Objects:*

- Encapsulate certain states.
- Communicate via message passing.
- Have methods corresponding to operations that may be performed on the corresponding state.

### *I.2.5.3.2. Similarities between agents and objects:*

- Both accept the principle of encapsulation and information hiding.
- Both recognize the importance of interaction.

### *I.2.5.3.3. Differences between agents and objects:*

*Table 1: Differences between agents and objects*

| Agent | Object |
|---|---|
| Agents embody a stronger notion of autonomy over their behavior (have control). | Objects are totally obedient to one another and do not have autonomy over their choice of action. |
| Agents have the reactive, proactive and social behavior. They are active entities which can decide what to do next. | Objects must wait to be told what to do next. |
| Agent-based approach defines a much more complex relation between agents. | Object-oriented approach defines a relation by static inheritance hierarchies. |
| Agents are multiple, dynamic and flexible. An agent can be a spouse, an employee, a customer and a landowner. | In object-oriented design, once the object is created, it can never change the class it belongs to. |

## I.2.6.All about environment:

### I.2.6.1. Properties :

#### I.2.6.1.1. Accessible VS inaccessible:

- An accessible environment is one in which the agent can obtain, complete, accurate, update information about the environment's state.
- Most moderately complex environment (including for example the everyday physical world and the internet) are inaccessible.
- The more accessible the environment is the simplest it is to build agents to operate in it.

#### I.2.6.1.2.Deterministic VS non-deterministic:

- ***Deterministic environment***: whenever the agent chooses to perform an action, it knows exactly what the outcome of the action is (there is one outcome for every single action).
- ***Non-deterministic:*** there are multiple outcomes from one action and the one is going to result is unknown.
- The more deterministic the environment is the easier an agent can perform actions (either way it will be unpredictable).

#### I.2.6.1.2.Episodic VS non-episodic:

- It relates to the kind of tasks that an agent carries out to an environment.
- A task in an environment is episodic if it's constructed of a series of distracted episodes, that means if one sub-task fails it won't affect other tasks. Otherwise it's non-episodic.
- Episodic is easier to the agent to make decisions about what action to preform next based only on the current episode it needs).

#### I.2.6.1.3. Static VS non-static:

- ***Static:*** if the agent is the only one operating on it, so it is predictable.

- ***Dynamic:*** several agents are operating within the environment, things change in ways beyond our control and we don't know how the environment is going to be.

## *I.2.7. Agents as an intentional system:*

Talking about agents as if they have cognitive state, in physical world we attribute feelings and desires to humans (we talk as if they have those feelings and desires). Predicting and explaining the behavior of rational agents in real world and assuming that there are going to interact and try to accomplish those desires giving that they have those beliefs.

### I.2.7.1. Dennett on intentional systems:

**Daniel Dennett(1987)[5]** coined the term intentional system to describe entities whose behavior can be predicted by the method of attributing beliefs, desires and rational acumen:

"A first order intentional system has beliefs and desires, but no beliefs and desires about beliefs and desires."

A second order intentional system is more sophisticated: it has beliefs and desires (and no doubt other intentional states ) about beliefs and desires (and other intentional states, both those of others and its own )."

### I.2.7.2. Can we apply intentional systems to machine?

- Ascribing beliefs to machines is legitimate when it expresses the same information about another machine that expresses about a person (a test for whether or not a person has a desire if he does, we expect that he acts so he achieves that desire).

- If it helps to predict the behavior of a computer processor, it allows us to predict the behavior of computer processor without knowing about how it's cooperating.

- There are multiple possible descriptions of the behavior of a system, example: when we try to know what a computer is doing, we use human language to describe it, so we talk about it as if it has desires.

### I.2.7.3. What can be described with the intentional?

The more we know about a system, the less we need to rely on animistic, intentional explanations of its behavior, but with complex systems a mechanistic explanation of its behavior may not be practicable.

As computer systems become ever more complex we need more powerful abstractions and metaphors to explain their operation, low-level explanations become impractical as the intentional stance is such an abstraction.

## *I.2.8. A formal model of agent and environment:*

- Abstract architecture for agent:

- **E={e,é,….}:**a model environment states; a set of all possible configurations of the environment.

- **Ac={α,ά,….}:**agents are assumed to have a repertoire of possible actions available to them, which transform the state of the environment. .

- A run of an agent in an environment is a sequence of an interleaved environments states and actions.

$$R=(e0,\alpha 0)\rightarrow(e1,\alpha 1)\rightarrow(e2,\alpha 2)\rightarrow(e3,\alpha 3)\rightarrow\ldots\ldots\alpha n\text{-}1\rightarrow en$$

### I.2.8.1. Runs:

let:

- **R**be the set of all such possible fine sequence (over E and Ac).
- **R Ac:**be the sub-set of those that ends with actions.

- **R E:** be the sub-set of those that ends with an environment state.

### I.2.8.2. Environment:

- A state transformer function represents behavior of the environment: **$\tau = RAc \rightarrow \phi(E)$.**
- Note that environments are: history dependent, non-deterministic.
- If **$\tau(r) = \emptyset$** there are no possible successor states to **R**. some say the run has ended “Game Over”.
- An environment **$Env = \{E, e_0, \tau\}$** where **E** is the set of environment states **$e_0 \in E$** is the initial state, and **$\tau$** is the state transform function.

### I.2.8.3. Agent:

- Agent is a function which maps runs to actions **$Ag: RE \rightarrow Ac$** (a function which takes as an input a run with the last thing happened when the environment changes state and gives as output a single action to perform).
- Thus an agent makes a decision about what action to perform based on the history of the system that it has witnessed to date.
- **$A\phi$**: be the set of all agents.

### I.2.8.4. *Systems:*

- A system is a pair containing an agent and an environment.
- Any system will have association of the set of runs of an agent **Ag** in an environment Env by **$R(Ag, Env)$.**
- Assume **$R(Ag, Env)$** contains only runs that have ended .

## *I.2.9. Perception, action and state:*

### I.2.9.1. Purely reactive agent:

Some agents decide what to do without reference to their history they base their decision-making entirely on the present, with no reference at all to the past. They just respond to their environment so they are not attempting to do any reasoning about what action to do. (it's a lookup table: "environment states → corresponding action") .

- We call such agent purely reactive : action**E→Ac**.

- A thermostat is a purely reactive agent:

**Action(e):** off if **e** = temperature OK

on otherwise}.

**Perception:**

- Getting perceptional data about the environment by sensors .

- The "see" function is the agent's ability to observe its environment whereas the "action" function represents the agent's decision-making process.

- Output of the "see" function is a perception: See:**E→Per**.

- Action: **Per*→Ac**take in consideration all the perceptions that it has received than what action should be performed.

- Sequence of perceptions → what action to perform.

- Which maps sequences of perceptions to act.

Figure 3: A model of an agent and its environment

### I.2.9.2. Agent with state:

We now consider agents that maintain state:

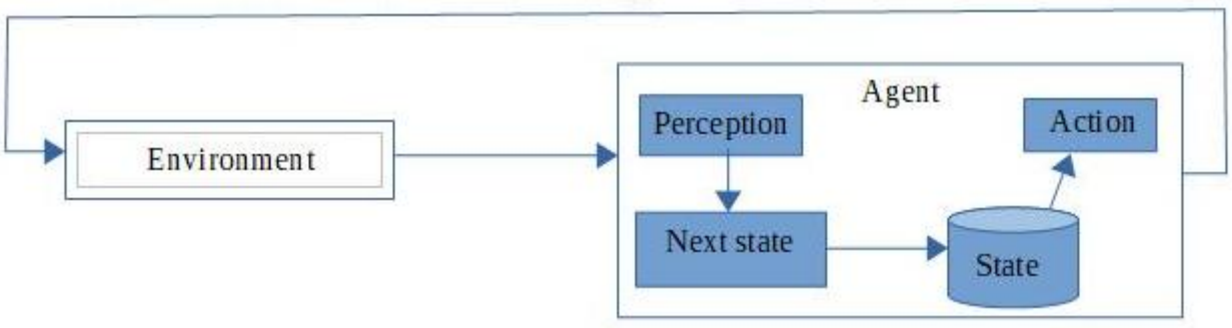


Figure 4: A model of an agent with its states

- The agent has an internal state which can update every decision.
- Next State function: is deciding what to record about the state of the environment.

**Perception:**

These agents have the same internal data structure which is typically used to record information about the environment state and history; let it be the set of all internal states of the agent.

**Next state function:**

A function "Next" is introduced, which maps an internal state and perception to an internal state.

**Next: $I \times Per \rightarrow I'$.**

I: an internal state.

Per: perception from environment.

**Action:**

The action selection function "action" is non defined as a mapping: **I→Ac** from internal to actions.

### I.2.9.3. Agent control loop:

The agent starts in some initial station.

- Repeat forever:
  - Observe environment state and generate a perception through: see().
  - Update internal state via "Next" function.
  - Select action via: Action().
  - Perform action.

## *I.2.10. Deduction reasoning agents:*

### I.2.10.1. Agents' *architectures:*

It is a software designed for agents, intended to support decision-making with the proprieties. It defines:

- Key data structure.
- On data structures.
- Control flow between operations.

#### *I.2.10.1.1. Pattie, 1991 [6]:*

A particular architecture methodology for building "agents"; it specifies how the agent can be decomposed into the construction of a set of component modules and how these modules and their interactions have to provide an answer to the question of how the sensor data and the current internal state of the agent determine the action and the future internal state of the agent. An architecture encompasses techniques and algorithms that support this methodology.

#### *I.2.10.1.2. Leslie, 1998 [7]:*

A Specific collection of software (or hardware) modules typically designated by boxes with arrows indicating the data and control flow among the modules. A more abstract view of an architecture is as a general methodology for designing particular modular decomposition for particular task.

### I.2.10.2. Types of agents:

#### *I.2.10.2.1. Symbolic reasoning agents (1956 – Present):*

It's the purest expression proposes that agents use explicit logical reasoning in order to decide what to do. Those who have dominated in the early days of AI are about to build agents that have symbolic and typically logical representation of their environment and they decide what to do via logical reasoning (practical reasoning agents).

#### *I.2.10.2.2. Reactive agents(1985 – Present):*

Problems with symbolic reasoning led to the reactive agents' movement. Those are behavioral agents where they're more responding to their environment rather than explicitly reasoning about it.

#### *I.2.10.2.3. Hybrid agents (1990 – present):*

Hybrid architectures attempt to combine the best of symbolic and reactive architectures, it's a mix between the past two types.

- Representing the environment symbolically:

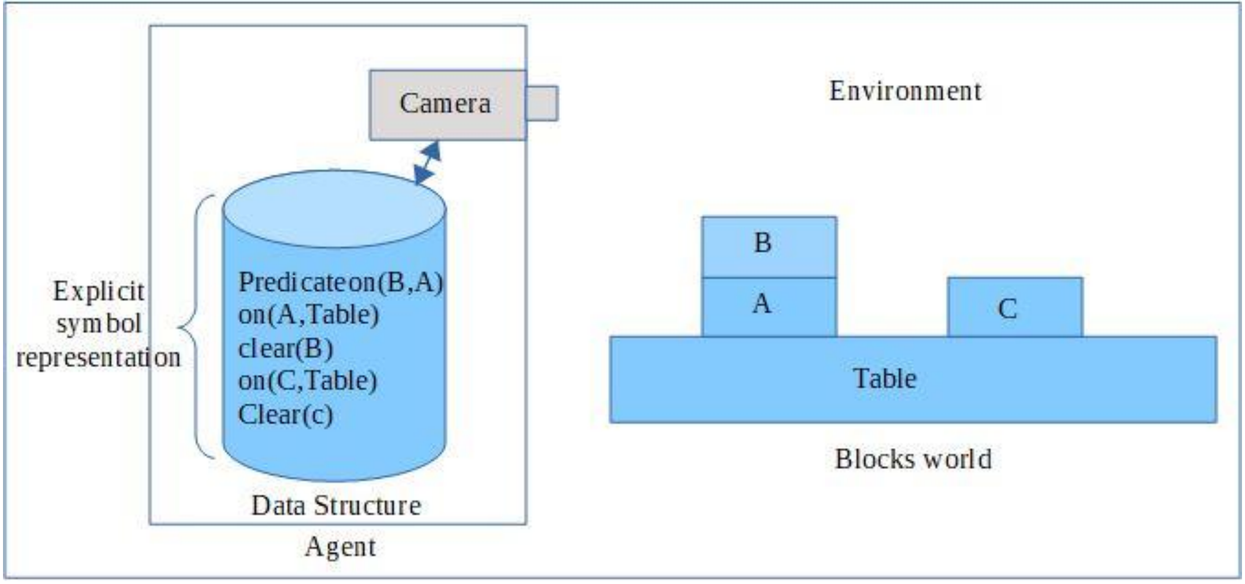


Figure 5: : Logical representation which captures the environment "describing in logical way using predicates"

### I.2.10.3. Interaction between agents:

"An interaction is the dynamic linking of two or more agents through a set of reciprocal actions ... Interactions are not only the consequence of actions carried out by several agents at the same time, but also the necessary element for the constitution social organization" (Ferber).

#### *I.2.10.3.1. Forms of interactions:*

*Table 2: Forms of interactions for agents*

| *Direct interaction* | *Indirect interaction* |
|---|---|
| An agent communicates by sending messages asynchronously to another agent or a group of agents. | Communication is carried out through the environment. |
| Once an agent knows another one, it will be able to communicate with it. | Agents interact through an intermediate entity |
| Agents must share a common vocabulary and agreed-upon meanings | The intermediate entity supplies sepecific interaction mechanisms and |

| | |
|---|---|
| to describe a domain subject. | access rules. |

### I.2.10.3.2. Interaction and cooperation between agents:

A multi-agent system differs from a group of independent agents in that the agents interact in order to collectively carry out a task or achieve a particular goal together. Thus an agent can be aware of the fact that it is not alone in his environment, and that it can interact with other entities. Which leads us to the notion of interaction which is defined as follows: "An interaction is a dynamic relation between two or more agents through a set of reciprocal actions " (Ferber).

This interaction may or may not be cooperative. We can characterize the system by the type of coordination implemented which can range from zero cooperation (total antagonism) to full cooperation. Also, cooperative agents can change their goals to meet the needs of other agents in order to ensure better coordination between them. This can lead to high communication costs.

Each agent can be characterized by three dimensions: its *goals*, its *capacity* to achieve certain tasks and *resources* available to it.

The interactions of the agents of an MAS are motivated by the interdependence of the agents according to these three following dimensions:

- ✓ Their goals may or may not be compatible.
- ✓ Agents may desire resources that others have.
- ✓ An agent may have the necessary capacity for another agent to accomplish one of the latter's action plans.

The way in which the interactions are accomplished makes it possible to obtain various situations on the MAS level, the following figure illustrates the interaction of an agent with other agents as well as with its environment.

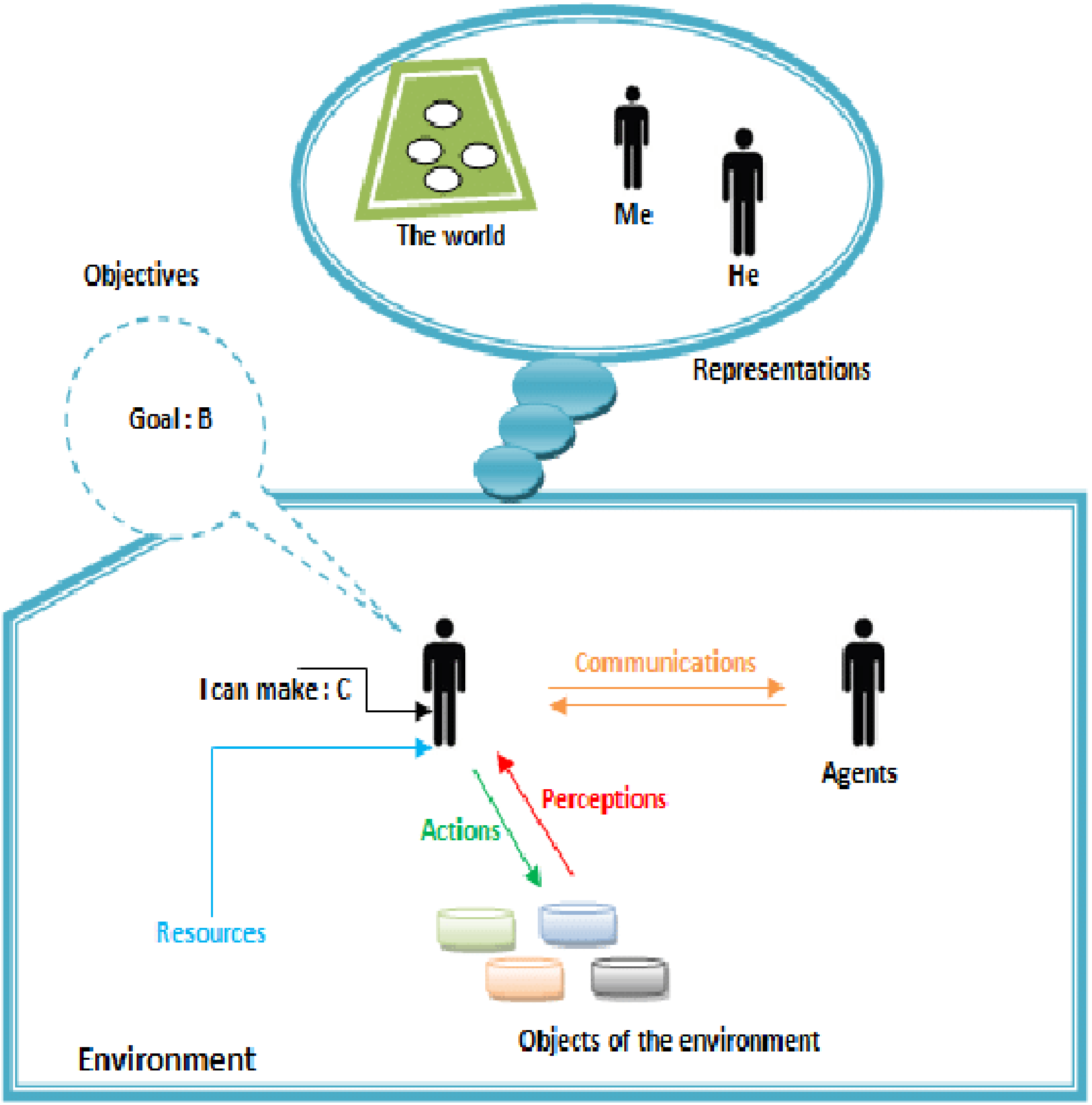


**Figure 6: Interaction of an agent with other agents as well as with its environment.**

### I.2.10.3.3. Interaction patterns:

- ✓ ***Collaboration:*** A way of distributing work between several agents.
- ✓ ***Coordination of actions*****:** How the actions of different agents must be organized in time and space
- ✓ ***Cooperation:***

- General form of interaction for agents capable of having an explicit project.

- Who does what, when, where, and with what means.

- Negotiation techniques are used to limit the effects of conflicts that appear.

***Cooperation = collaboration + coordination + conflict resolution***

## I.2.11. Organizations:

MAS can be organized:

- ✓ A priori: in a hierarchical structure, of walking, of community, of company (Grislin, 95) [8], (Mandiau, 99) [9].
- ✓ By emergence: the structure of the organization appears following interactions between agents.

## I.2.12. Example of Application areas of MAS:

The applications of multi-agent systems cover several fields. These include cooperative information systems, sociological simulation, tools, documentaries adapted to the Web, autonomous cooperative robots, video games (multiplayer), distributed problem solving, etc.

### *Simulations:*

The objective of which is the modeling of real world phenomena, in order to observe, understand and explain their behavior and evolution.

MAS quickly found an extremely favorable field for their development in the field of modeling complex systems such as human and social science along with the living one. Multi-agent simulation makes it possible to test quickly change some assumptions; it also allows to integrate new agents and to edit, on a practical level, the results to compare the experiments to each other, moreover it allows to preserve the heterogeneity of the system to be simulated.

## *I.3. Conclusion:*

Through this chapter we have given brief definitions about agents and multi-agent systems along with explanation of concepts like the environment, reasoning, interaction and communication.

The next chapter is devoted to multi-robots' systems and swarm robotics.

# *CHAPTER 02 :*

# *MULTI-ROBOTS' SYSTEM & SWARM ROBOTICS*

# II. CHAPTER 02: MULTI-ROBOTS' SYSTEMS AND SWARM ROBOTICS:

## *II.1. Introduction:*

The study of multi-robot systems -in the field of mobile robotics-, has grown significantly in size and importance in recent years. Having made great progress in the development of the basic problems concerning single-robot control, many researchers shifted their focus to the study of multi-robot coordination and cooperation. Since application domains and tasks constantly keep challengingMRS (Multi Robot System) are of increasing complexity so the robots'cooperation can be regarded as a fundamental and crucial features.

The word *robotics* first appeared in *Isaac Asimov*'s science-fiction story *Runaround* (1942). Along with Asimov's later robot stories, it set a new standard of plausibility about the likely difficulty of developing intelligent robots and the technical and social problems that might result.

In thischapter, we speak about the forms of cooperation and coordination realized in the MRS and what they lead to and an important related concept which is Swarm robotics.

## *II.2. Multi-robots' Systems:*

### II.2.1. Robots:

#### *II.2.1.1. Definition:*

A robot is a machine designed to physically act on its environment execute one or more complex tasks automatically with speed and precision in order to achieve a certain and known goal. Performing tasks correctly in a durable time, requires having functions of perception, decision and action even when the robot encounters new unexpected situations.

Programming a robot allows it to manipulate objects by performing various movements, it consists, first of all, in specifying the sequence of movements it will have to perform. Some robots are endowed with "senses"; that is to say of a more or less important set of measuring and assessment instruments like (camera, thermometer, rangefinder, ...) allowing the robot program to decide on the movement best suited to external conditions.

### *II.2.1.2. Types of robots:*

Generally, there are five types of robots:

#### II.2.1.2.1. Pre-Programmed Robots:

Pre-programmed robots operate in a controlled environment where they do simple, monotonous tasks. An example of a pre-programmed robot would be a mechanical arm on an automotive assembly line. The arm serves one function — to weld a door on, to insert a certain part into the engine, etc. — and its job is to perform that task longer, faster and more efficiently than a human.

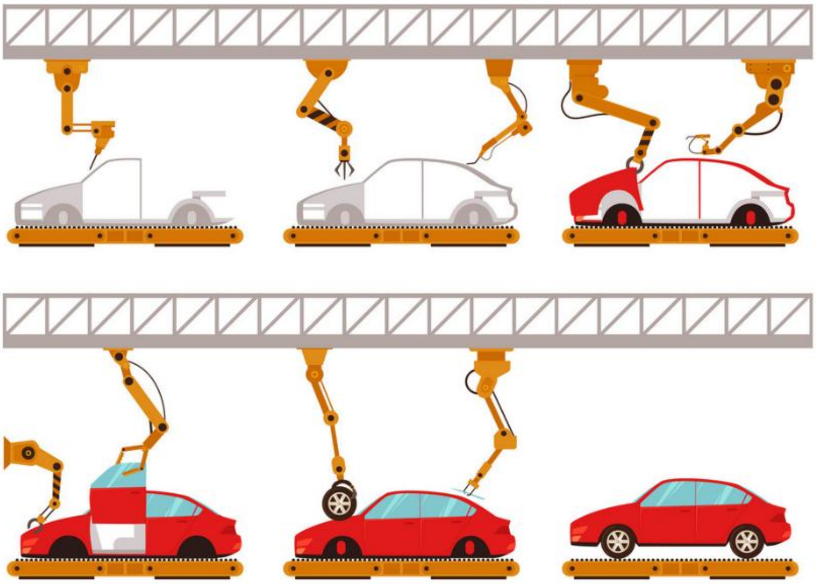


**Figure 7: Car assembly line with mechanical arms**

### II.2.1.2.2. Humanoid Robots:

Humanoid robots are robots that look like and/or mimic human behavior. These robots usually perform human-like activities (like running, jumping and carrying objects), and are sometimes designed to look like us, even having human faces and expressions. Two of the most prominent examples of humanoid robots are Hanson Robotics' Sophia [10] andSoftBank Robotics' Pepper .

Figure 8: Sophia (Hanson Robotics)

Figure 9: Pepper Soft(Bank Robotics)

### II.2.1.2.3. Autonomous Robots:

Autonomous robots operate independently of human operators. These robots are usually designed to carry out tasks in open environments that do not require human supervision. An example of an autonomous robot would be the Roomba vacuum cleaner, which uses sensors to roam throughout a home freely.

**Figure 10: Roomba vacuum cleaner**

### II.2.1.2.4. Tele-operated Robots:

Tele-operated robots are mechanical bots controlled by humans. These robots usually work in extreme geographical conditions, weather, circumstances, etc. Examples of tele-operated robots are the human-controlled submarines used to fix underwater pipe leaks during the BP oil spill or drones used to detect landmines on a battlefield.(See figure 11,12).

Figure 11: A drone that can maps, detect and detonate landmines

Figure 12: A submarine tele-operated robot used  on marine biology

### II.2.1.2.5. Augmenting Robots:

Augmenting robots either enhance current human capabilities or replace the capabilities a human may have lost. Some examples of augmenting robots are robotic prosthetic limbs or exoskeletons used to lift hefty weights.

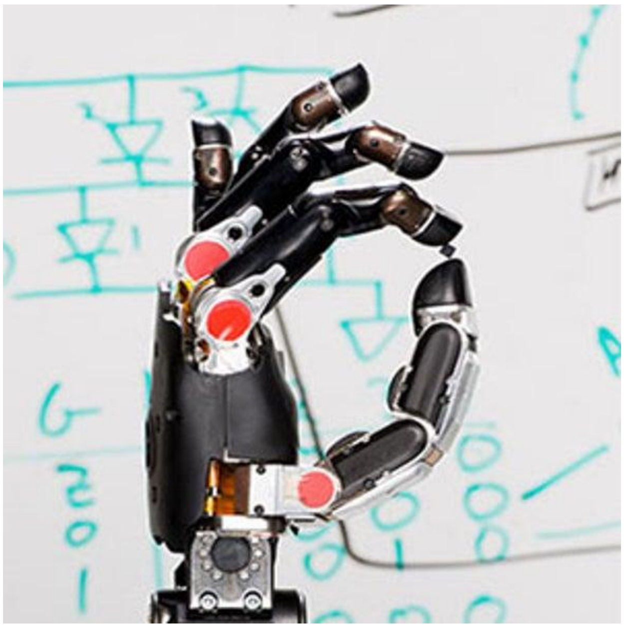


Figure 13: Robotic prosthetic limbs

### II.2.1.3. Uses of robots:

- Helping fight forest fires.
- Working alongside humans in manufacturing plants (known as co-bots).
- Robots that offer companionship to elderly individuals.
- Surgical assistants.
- Last-mile package and food order delivery.
- Autonomous household robots that carry out tasks like vacuuming and mowing the grass.
- Assisting with finding items and carrying them throughout warehouses.
- Used during search-and-rescue missions after natural disasters.
- Landmine detectors in war zones.

### II.2.1.4. Robots' navigation:

According to Levitt and Lawton (1990) [11], navigation is defined as the process to answer the following three questions:

a. "Where am I?" (Localization).

b. "Where are other places with respect to me?"
c. "How do I move and get to other places with respect to me?"

Navigationis different from other forms of spatial behavior such asexploration, or foraging, in that there is an explicit referenceto a goal location (Franz & Mallot, 2000) [12].It is basically aset of techniques that allow you to:

- Know the position (coordinates) of a mobile robot in relation to a system of reference, or with respect to a determined fixed point.
- Calculate or measure the route to follow to reach another known coordinate point while respecting a certain number of constraints and criteria resulting from several factors, which generally depend on the characteristics of the robot, the environment, and the type of task to be performed.
- Calculate any other information relating to the movement of this mobile (distance and duration, speed of movement, estimated time of arrival, etc.).

## II.2.2. Multi-robot Systems:

### *II.2.2.1. What is a multi-robot system?*

A Multi-Robot System can be characterized as a set of robots operating in the same environment. However, robotic systems may range from simple sensors, acquiring and processing data, to complex human-like machines, capable of interacting with the environment in fairly complex ways.

Robots of the MRS can be homogeneous or heterogeneous, mobile or fixed, they are able to communicate with each other and cooperate to improve the efficiency of the execution of tasks or to allow to perform tasks that cannot be performed individually.

### II.2.2.2. Why a multi-robot System?

A multi-robot system can often deal with tasks that are difficult or complex or even impossible to accomplished by an individual robot. A team of robots provides some redundancy and contributes cooperatively to perform the assigned task in a faster way beyond what is possible with individual robots.

Other advantages:

- ✓ ***Robustness:*** "If one robot fails, the others step in." => The performance in a MRS is not gravely affected in case of a robot's failure.
- ✓ ***Performance:*** "More robots could get a task done faster!" => To obtain a high-leveled performance, a lot of robots interfere to solve a problem.
- ✓ ***Flexibility:*** it is possible to perform the assigned tasks in various ways. This is mainly induced by the redundancy of robotic entities.
- ✓ ***Scalability:*** "If the problem gets bigger, just get more robots" => The complexity of the task to be performed is too high for a single robot so it requires cooperation of a group of robots.
- ✓ ***Specialization:*** "While some robots do this, other are already doing that" => for better performance, several tasks are executed in parallel.

## II.3. SWARM ROBOTICS:

Direct communication in the organization of artificial agents is used in simple tasks such as signaling agents' preferences and states. This was inspired by the emergence of meaning found in natural languages because if the communications skills were more complex the design of robot swarm would be more autonomic and adaptive.

In this part of the chapter we get to explain how to emerge inside a population of autonomic artificial agents a conventional lexicon to specify objects in the real world, and this is known as "Language games" inspiedby offline learning [13]. Then we move forward to the communication where the words of the emerged lexicon refer to shared meanings when the different swarm robots cannot perceive nor categorize the world in the same way.

## II.3.1. Definition:

A swarm of robots is a group of robots in motion or turmoil. Swarm robotics is an approach to the study of Multi-robots' systems that aims at designing complex collective behaviors using simple robots.

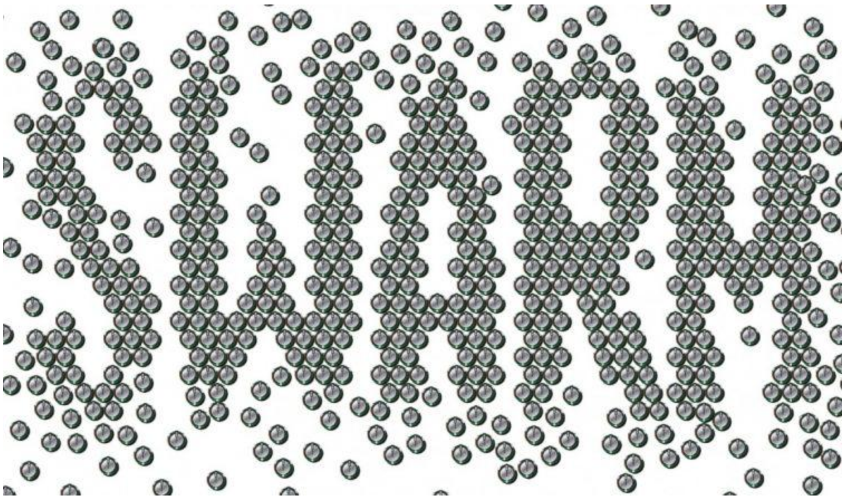


Figure 14: Illustration of a swarm

## II.3.2. Constraints to build a lexicon in a population of robots:

### *II.3.2.1. Locality:*

- Robots are distributed.
- No robot could have a complete clue about the behaviors of all the other robots in the population.
- The lexicon of the population is not directly accessible by robots.

### *II.3.2.2. Non-telepathy:*

A robot cannot get direct access to other robots' data structures whether to inspect or modify them. It's incapable of telepathy.

## II.3.3. Forms of interaction [14]:

Table 3: Forms of interaction in communication

| Indirect communication | Direct interactions | Direct communication |
|---|---|---|
| **-Like in insect societies,**<br>**- It is resulted from modifications of the environment.**<br>**- eg: Ants depositing pheromones during their foraging trips.** | -Physical contact<br>-Provokes a response on the receiver<br>-eg: Pushing/pulling | -Concurrent exchange of information among individuals without physical contact<br>-Ubiquitous modality of interaction in swarm robotics. |
| **-Require much more specific sensors and actuators**<br>**-Eg: UV-light emitters coupled with a bespoke floor material or force/torque sensors.** | | -Implementable with Radio links, infra-red transceivers and visual signals<br>-Eg: colored LEDs |

Rules of communication are designed specifically for the task at hand which limits the swarm's autonomy, like for example: Self-assembly and coordinated motion; robots can negotiate obstacles during exploration of complex, unstructured and uncertain environment but if the conditions of self-assembly are predefined, no adaptation is possible for there will be different and new obstacles in heterogeneous environments. Richer ability of communication allows swarms to negotiate better which leads to autonomous and self-organized formation of different categories for contexts that requires different responses.

Robots interact and negotiate to achieve online adaptivity using a mechanism represented by Language games.

## II.3.4.Evolution of communication and language:

The main challenge is to determine the signal and the response to it that concurrently assures the evolution of a functional communication.

### *Experiment 01: SIGNALING SYSTEMS*

- Small colonies of robots were evolved within a particular scenario did not especially encourages communication (Floreano, 2007) [15].
- Robots were assigned a foraging task which to find food sources in an environment that also contains poison sources.
- At the end of an evolutionary optimization process, robot swarms with visual communication system performed the task better with respect to communication-less swarms.
- Two signals emerged in different populations: robots shared the position of the food source (attraction) or only signaled the poison source (repulsion).

*Conclusion*:

Signaling systems with evolved controller are supposed to produce more adaptable behaviors so that robots could recognize only after some exploration.

The current use of automatic design methods limits the adaptivity necessary for communication in uncertain environments because the evolution/emergence of communication is related strictly to the training step (the evolved rules remain identical after deployment).

### II.3.5. Language games:

According to Wittgenstein and the philosophy of language (1953) [16] Language games are simple abstractions of a real-world language. Language games can be defined to be played between robots with the purpose of mimicking real-world's dynamics leading to the emergence of language.

And we see in the up coming figure all the operations on this kind of naming games that must be done on 5 steps :

-First:a speker robot get a sence from his environment through sensors.

-second : the speaker robot generate a random desciption fro this sence.

-Third : the speker robot sends the description ande the sence to the hearer robot.

-Fourth : the hearer robot checks if the sence and its description is already in his inventory.

And here it has tow choices :

- If the word exist in the inventory:the game is a success .
- Otherwise it's a failure and in this case the hearer robot must :

  - Generate a random discription for the received sence and checks if it's the same as the description received.

- If (description received = description generated) -> Game is success
- Else Game is failure and it takes the Speakers description as a description for that sence fom the envionment and it updates his inventory

And the game continue for all the existing robots from the environment.

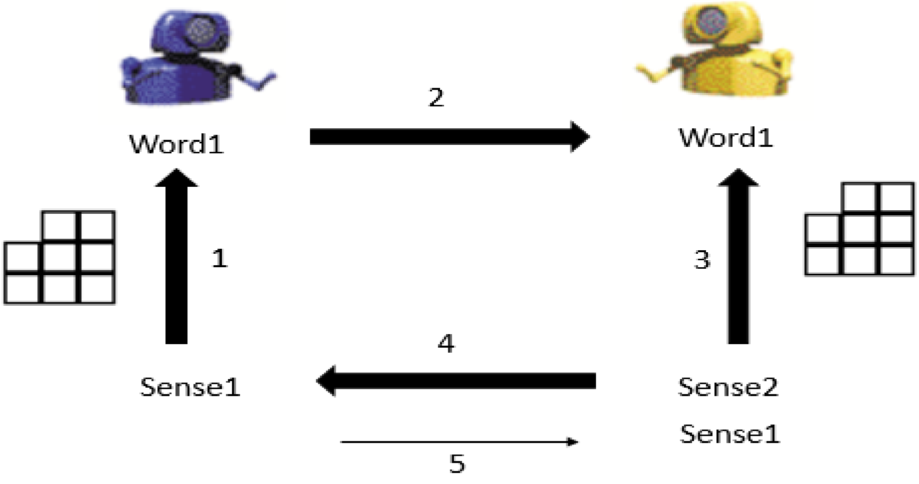


Figure 15: A model of "Language-Naming Games"

There are kinds of langage games:

### *II.3.5.1. The imitation game (De Boer, 2000) [17]:*

Vowels Vocalization or Vowel systems, the concept of it:

- Robots are equipped with an articulatory synthesizer, a module for calculating the distances between different vowels and a repertoire to store vowel prototypes.
- Two robots are selected to do it.
- The "initiator" selects a random vowel from its repertoire and utters it.
- The "imitator" tries to imitate the selected vowel by uttering the closest vowel in its own repertoire.
- The "initiator" has to find the closest vowel to the one uttered by the imitator in its own repertoire (find the initial vowel).
- Both robots either emerge their vowels or add a new one.

### *II.3.5.2. The guessing game (Steels, 2001) [18]:*

- The "speaker" chooses a concept with a context and communicates the corresponding word to the hearer.
- The "hearer" has to guess which topic was chosen based on the communicated word.
- In case of failure, both of them update their inner representation of the concept.
- Concept of Gavagai thought experiment: addressing the inscrutability of reference in a computational context, one word can never have exactly the same meaning for different agents [19].

### *II.3.5.3. Category game (Puglisi,2008. Baronchelli,2010) [20]:*

- Aims to self-organize discrete sub-intervals of one or many perceptual channels through negotiation dynamics.
- No pre-defined category to start with.

- A pattern of categories is developed by repeated interactions between agents.
- Eventually, a global agreement emerges within the population.
- Communication grounding: Assuring a matching signified/signifier link between words and concepts to be exploited in future communications.

### *II.3.5.4. Naming games (Steels, 1995, 2007) [21]:*

- The category game canbe simplified into the *naming game*.
- Categories are provided from the beginning.
- Focus on the negotiation dynamics and the emergence of an agreement.

### II.3.6. Language games for Robots' swarms:

- LGs and swarm robotics' combination enhance the efficiency and adaptivity of communication between robots in the swarms.
- It also provides new means to study the evolution of language.
- The main key is self-organization where agents display their behaviors resulting a never ending evolution with new words or concepts arising when needed due to constant changes in the environment.

### II.3.7. Degrees of coupling between LGs and swarm robotics:

1- The robot behavior is not affected from the LG played upon repeated encounters.

2- The LG affect the behavior of robots but the latter have no direct influence on the language evolution.

3- The behavior of robots affects the language evolution.

- Case 1 and 2: The language only carries information that is useful for the observer (provides descriptions of the environment).
- Case 3: the language carries an emergent semantics that are intrinsic to the system (Robots use language purposefully). Eg: Developing grounded symbols (Harnad, 1990) [20].

## II.3.8. Minimal naming games:

- Attempts to exploit the power of LGs for the coordination of robot swarms.
- Two or more robots interact to assign a unique name to a single object (like the imitation game).
- The "speaker" chooses one of the words in its inventory and broadcast it to all the other robots in range.
- The "hearer" checks the existence of the word in its inventory and updates it.
- In case of failure: the word is included into the inventory.
- In case of success: The word is maintained in the inventory and all other are excluded.

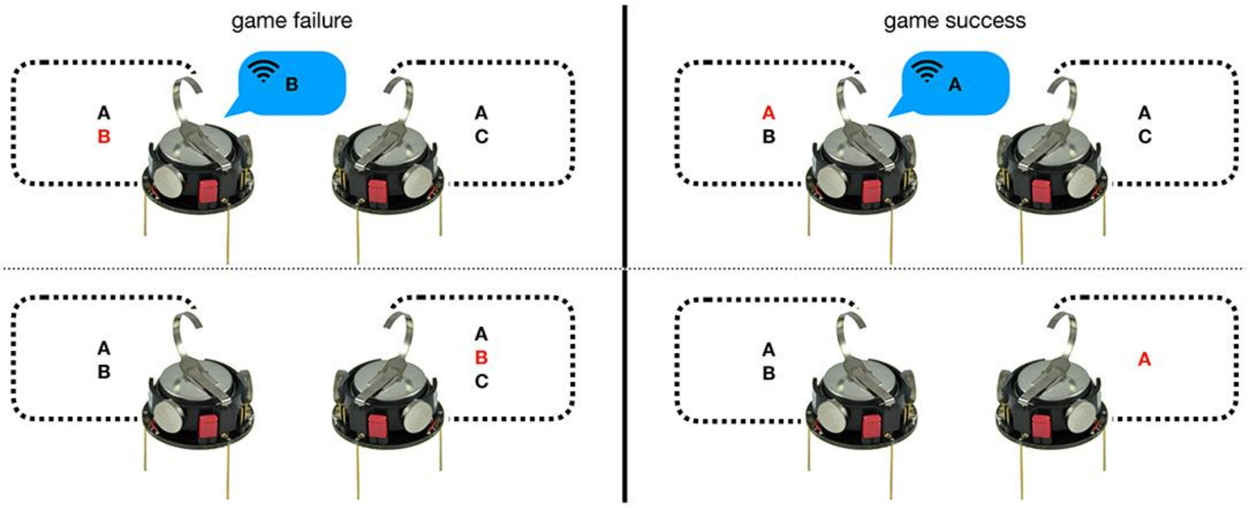


Figure 16: A model of "Minimal Naming Games"

## II.3.9. (Tirianni, 2016) [21]:

Focus is given to reach consensus on a single word for a single object within a population of communicating robots.

- No interaction between LGs and robot behaviors.
- Communication network is shaped by the encounters between the robots, each independently performing a simple random walk.

- Collisions between the wireless transmitted messages (due to the communication protocol and the density of robots) impacts the capacity of information transfer within the swarm but doesn't weaken its ability to reach consensus only with longer delays.

### II.3.10. (Miletich, 2019) [22]:

- From an embodied version of MNG.
- A possible correlation between a foraging task and the language dynamics.
- The topology of the interaction network is determined by the foraging task.
- Transition of time from a well-mixed population to a swarm polarized in two-sub populations with little contact.
- A robot creates a word when it discovers a source new to him (the robot's behavior influences the languages dynamics).

## *II.4. Conclusion:*

In this chapter, we have introduced robotics. Subsequently we focused on multi-robot systems and their areas of use as well as how robots can cooperate in such systems then we moved to swarm robotics and the emergance of language inside a poulation of autonomous robots. The following chapter is devoted to the conception.

# CHAPTER 03 :

# CONCEPTION

# III. CHAPTER 03: CONCEPTION

## *III.1. Introduction:*

A robot is supposed to replace humans to solve problems which can be quite complex sometimes. A group of robots collect objects inside an uncertain environment and interact with each other, but they don't use the same language. How could these robots emerge a single common lexicon to communicate with each other?

In this chapter, we choose the emergence of language in a cooperative multi-robot's system. We present the conception of our application using different language games' algorithms and UML's communication diagram.

Our application concerns the development of language and lexical categories inside a population of heterogeneous robots thus the communication between them using the emergent and developed lexicon.

## *III.2. Language games:*

### III.2.1. Environment:

- It consists of perceptions/meanings that agents could share.
- Perceptions P={$p_1$,$p_2$, …., $p_n$}, for our case the perceptions taken from the environment are: the name of the cherry, its position (represented with digits), its possible colors and sizes therefore; *P = {Cherry, Green, Blue, Red, Big, Little, Medium, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.*
- It allows the transmissions of words (descriptions) between the robots; D = {$name_1$, $name_2$, …, $name_n$}.

### III.2.2. *Agents:*

- Each agent has a shared memory/dictionary (empty at the beginning).

- The memory consists of associations/pairs of perceptions and descriptions R(perception, name)
- The relation R is characterized by a score and the latter increases when the relation is used successfully.
- An agent uses the memory to associate a perception to a description (codes it: perception => name) or finds the perception from the given description (decodes it: name => perception).
- After the language games the memory is filled with the relations with the highest scores.

## III.2.3. Interaction:

### *III.2.3.1. Progress:*

1) The speaker chooses a perception randomly from the list of perceptions describes it (gives it a name). If the description doesn't exist in its memory it systematically creates a new one.
2) The description is transmitted from the speaker to the hearer.
3) The hearer checks if the received description is in its inventory. If it doesn't exist, he creates w new relation between the received description and the perception indicated by the speaker.
4) The hearer indicates the found perception to the speaker non-verbally.
5) If they agree (same descriptions for the same perception), the game is a success.

### *III.2.3.2. Adaptation:*

- *Success:*

The hearer increase/increment the score of the relation (perception, description) by a unit.

- *Failure:*

- The hearer decrease/decrement the score of the relation (perception, description) by a unit.
- The speaker communicates with the hearer and sends it the perception he designated.
- The hearer then increments the score of the used relation.

The steps to develop a lexicon is as follows:

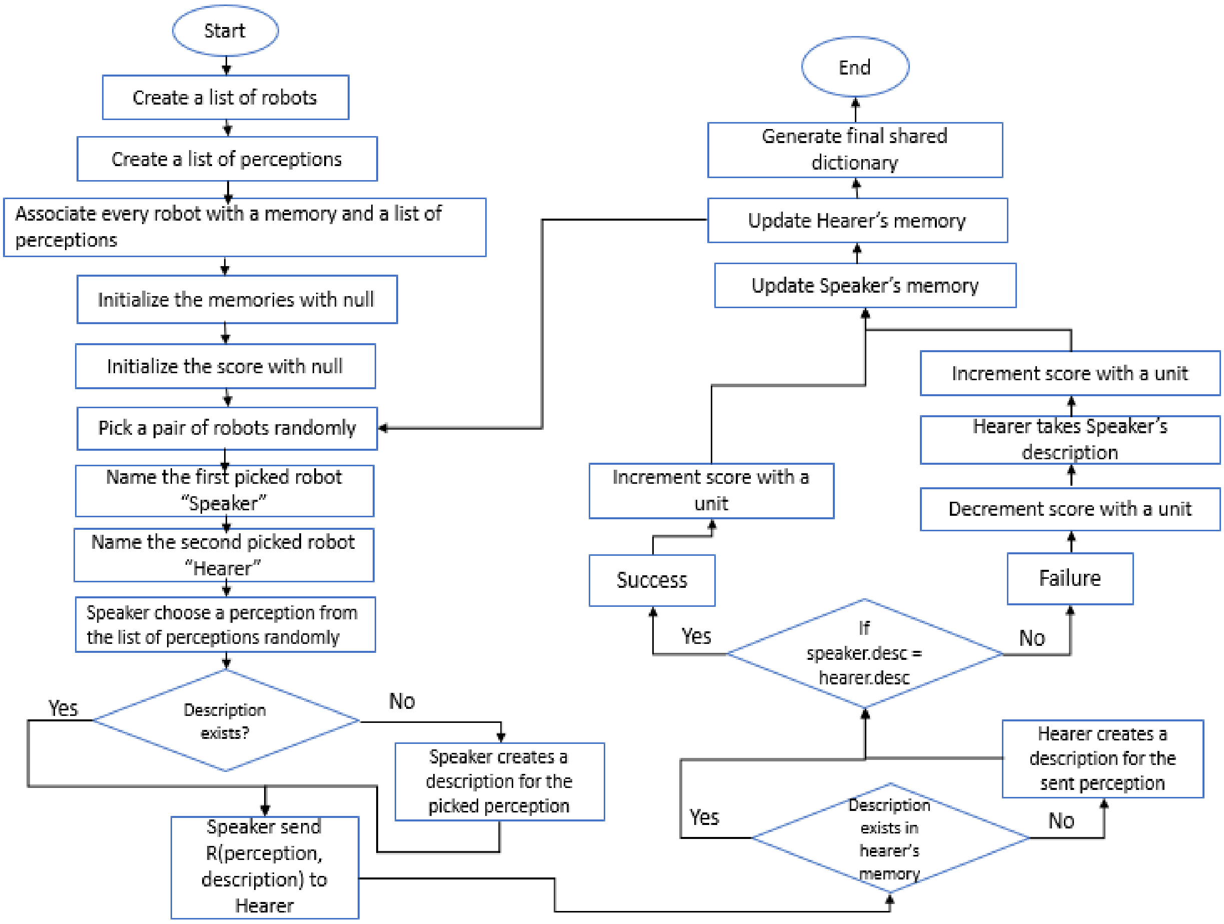


Figure 17: General organization chart of language games

## III.3. Multi-robots' system:

Our chosen MRS is composed of heterogeneous robots with several roles:

- ✓ ***Locator:*** this robot is responsible for locating the objects (cherries for our case).
- ✓ ***Transporter:*** this robotis responsible of picking the concerned cherries, because each category of cherries has a specific transporter. Eg:Black transporters are meant to pick small cherries.
- ✓ ***Coordinator:*** this robot isresponsible for ensuring the coordination between the various robots in the system.

The first two types of robots are mobile robots which can be seen as a autonomous entities which perceive their environment and act on it.

To do this, the robot navigation is explained in the next section.

### III.3.1. Navigation:

In order to allow a robot to reach a desired position (in our case the position of cherries), there are two main families of methods.

1) The methods without explicit trajectory (fields of potentials, neural networks and fuzzy logic) which instead seek to control global movement of the robot so as to guide it to its goal.
2) The other family, that of trajectory tracking methods, allows the robot to follow a given reference trajectory "as best as possible", knowing the kinematic constraints (Morette, 2009) [23].

We previously opted for the method without an explicit trajectory.And this, by making obstacles repellent for the robot in order to move away from it and keep a safe distance using bumpers and sensors, and on the other hand give back the rest of the attractive environment (void) for him to navigate.this will allow our robots to navigate freely in their environment .

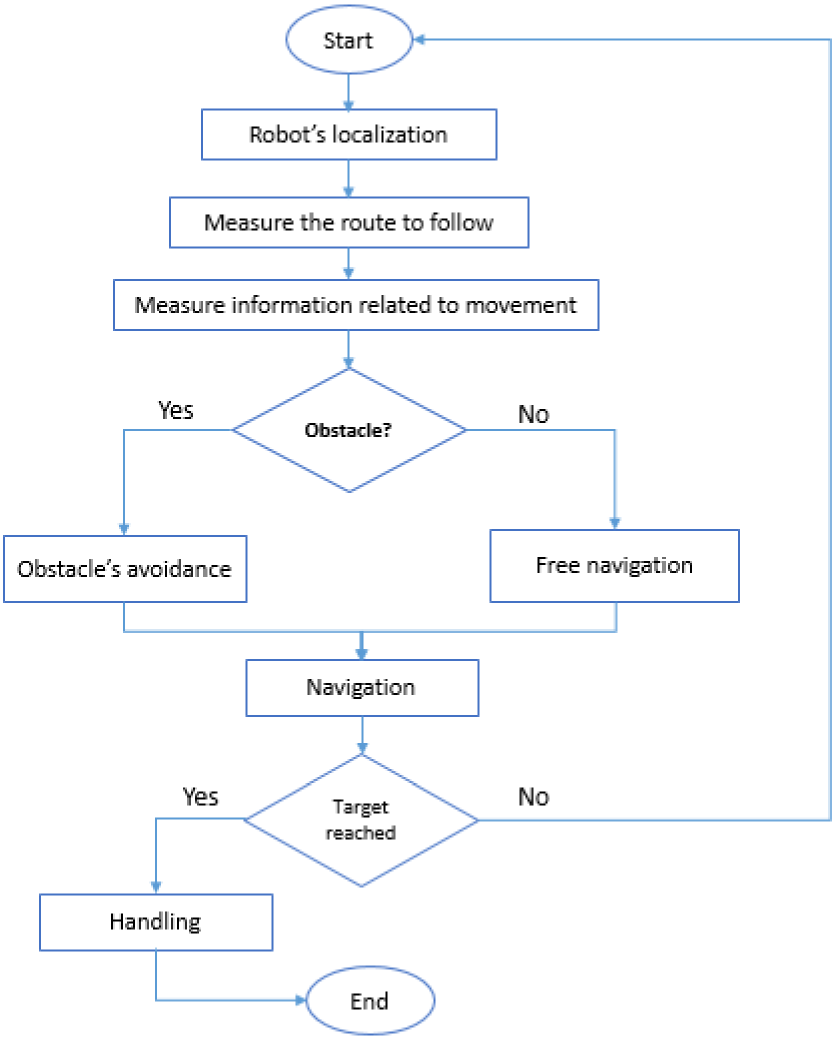


**Figure 18: General organization chart of robots' navigation**

### III.3.2. Environment's description:

- ✓ The environment: the navigation space is full of cherries.
- ✓ Robots: in our case there are types of robots:

1. *Locator robot:* it has a position and a name.
1. *Coordinator robot:* the robot that links transporters with locators.
2. *Transporter robot:* every transporter has a name, position, a color according to the size of the cherry.

- Objects: there are two types of objects: obstacles and cherries.

- Obstacles: the environments’ walls.

- Cherries: every cherry has a name, a color, a size, a position and an action.

### III.3.2.1. Locator robot:

It is capable of navigating inside the environment, once it finds a cherry it returns the descriptions of its’ characteristics (size, color and position) and send them to the coordinator robot to treat them.

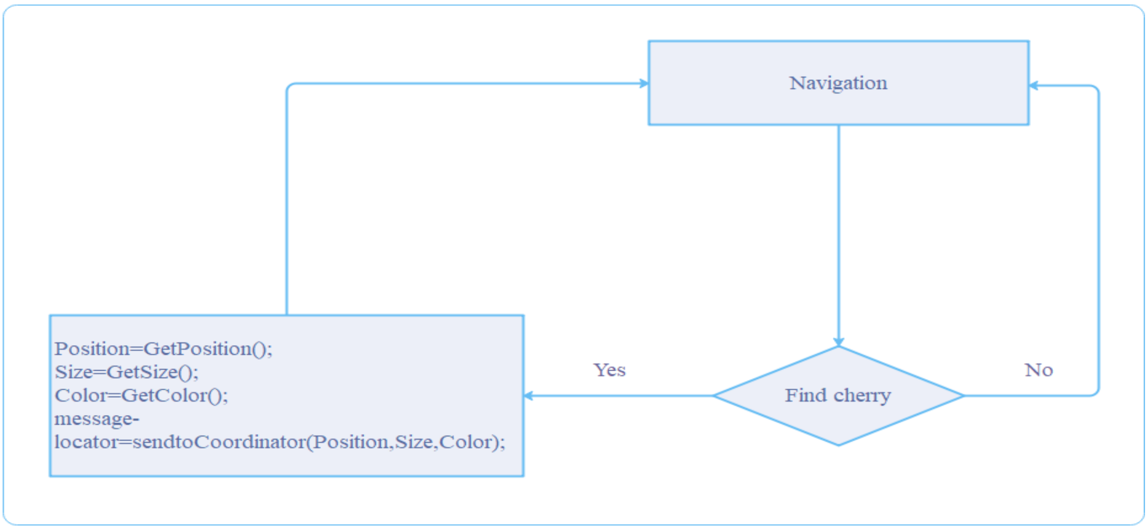


Figure 19: Organization chart of a locator robot

### III.3.2.2. Coordinator robot:

It receives the descriptions from the locator robot, decodes them and selects the corresponding transporter according to the received descriptions from the locator robot. Eg. The red transporter is selected to transport big cherries.

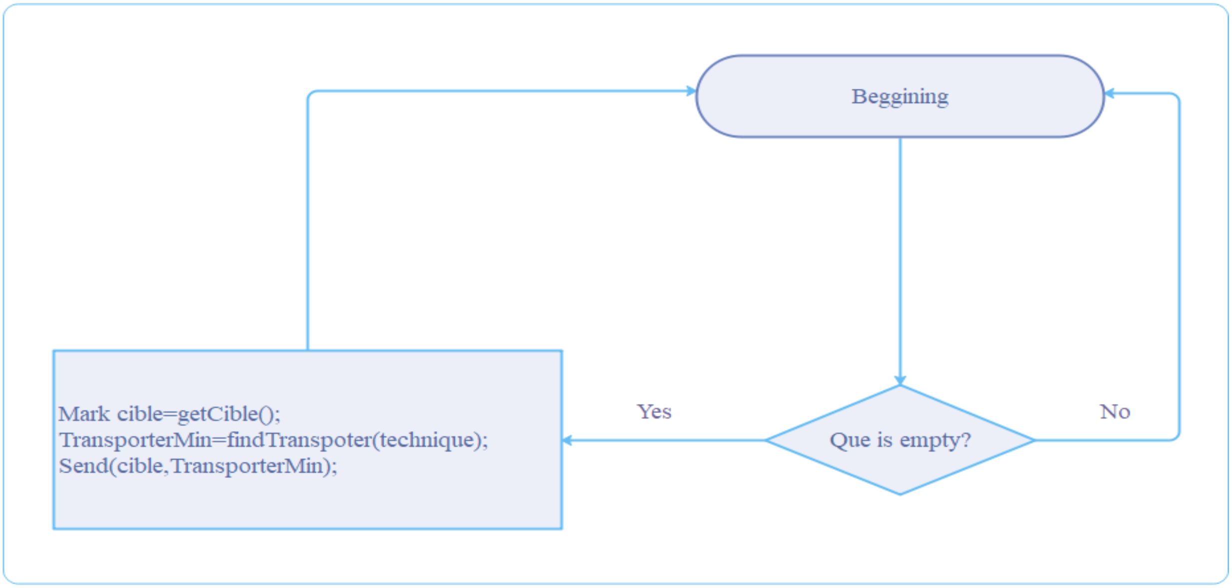


Figure 20: Organization chart of a coordinator robot

### *III.3.2.3. Transporter robot:*

It has to decode the descriptions sent by the coordinator robot and send back the corresponding perception's name in a sign that it understands what it is told to do and goes to the specified cherry to detach it from the environment.

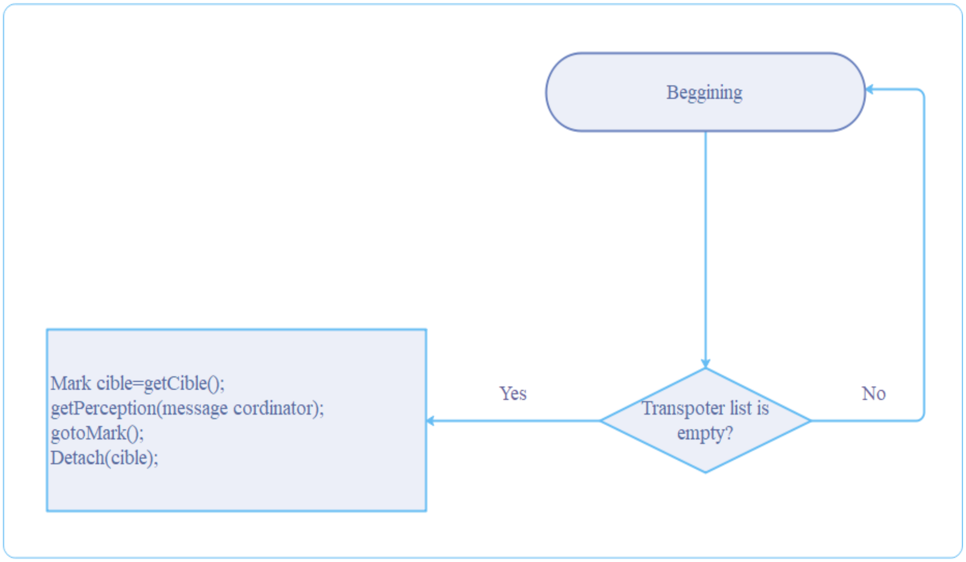


Figure 21: Organization chart of a transporter robot

The next figure represents the sequence diagram of the system. It is the graphical representation of the interactions between the system and its actors in a chronological order according to UML formulations, in our case it represents the interactions between the robots.

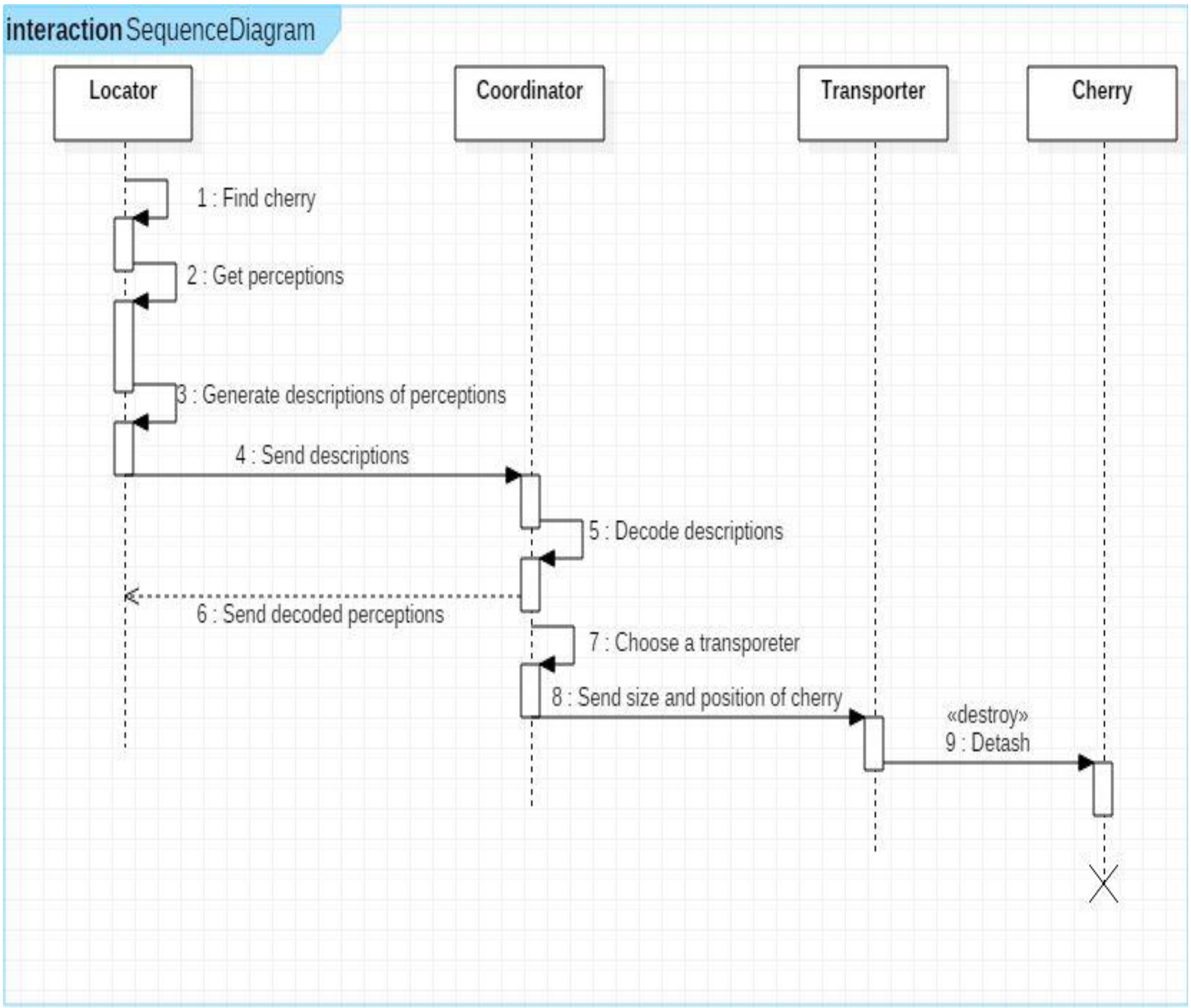


Figure 22: Sequence diagram of the system

## III.4. Conclusion :

In this chapter, we tried to illustrate and clarify the different functionalities of our system and every part of it. we explained the strategy our work is based on and how the communication happens.

# CHAPTER 04 :

# IMPLEMENTATION

# IV. CHAPTER 04: IMPLEMENTATION

## *IV.1. Introduction:*

After we have finished with the conception of our system, we now move on to the third activity of the development process, which is the implementation phase that we summarize in this chapter.

The chapter is composed of two main parts. First, we present the tools and technologies used to implement our system. Finally, in the last part, we present the main features of our application and its architecture in presenting screenshots of the several interfaces used in it.

## *IV.2. Environment and technologies used in the development:*

This part is devoted to the presentation of the different technologies that have served us tools throughout the development of our application.

### IV.2.1. Development environment:

We present in the following the realization of the application as well as the tools that were used during the design and implementation phase.

In computer programming, a development environment is a set of tools to increase the productivity of programmers who develop software. Some environments are dedicated to a particular programming language.

#### *IV.2.1.1. Modeling:*

***UML:*** (Unified Modeling Language) is a universal modeling language, graphical and textual which allows to present and manipulate object concepts with a considerable precision thanks to the various diagrams it offers. It facilitates understanding complex abstract representations.

***Edraw Max***: is an all-in-one diagram-making tool that can be used for flowcharts, mind maps, fishbone, network and UML diagrams, floor plans, office layouts, Gantt charts, business cards and flyers, wireframes, infographics, and presentations. We used it to create the organigrams [24].

***Enterprise Architect:*** Is a UML modeling and design software, published by the Australian company SparxSystems. Covering, by its functionalities, all the stages of the application design cycle, it is one of the most recognized design and modeling software. We used it to create the sequence diagram.

### IV.2.1.2. Implementation:

***Java:*****Java** is an object-oriented programming language developed by James Gosling and colleagues at Sun Microsystems in the early 1990s. Unlike conventional languages which are generally designed either to be compiled to native (machine) code, or to be interpreted from source code at runtime, Java is intended to be compiled to a bytecode, which is then run (generally using JIT compilation) by a Java Virtual Machine. [26]

***Netbeans IDE:*** it is an open source Integrated Development Environment (IDE) designed by Sun Microsystems. Written in XML, Java. NetBeans supports various languages of programming like java, PhP, C / C ++ and so on. It provides all facilities of a modern IDE (graphical interface editor, refactoring, web page editor and color editor). It is available on Windows, Linux, Solaris and Mac OS X, it is independent of operating systems and constitutes a rich platform that allows development of applications on servers (web and JEE applications), mobiles and XML Web Services. [27]

***Scene builder:***JavaFX Scene Builder is a visual layout tool that lets users quickly design JavaFX application user interfaces, without coding. Users can drag and drop UI components to a work area, modify their properties, apply style sheets, and the FXML code for the layout that they are creating is automatically generated in the background. The result is an FXML file that can then be combined with a Java project by binding the UI to the application's logic. [28]

***Simbad:*** It is a 3D Java robot simulator with scientific and educational objectives. Being open source and free, it is mainly dedicated to researchers/programmers who want a simple basis for studying artificial intelligence, machine learning and more generally AI algorithms, as part of the autonomous robotics and autonomous agents. It provides multiple features such as 3D visualization and detection, sensors (sonars), contact (bumpers) and visuals (monoscopic color camera), as well as User Interfaces for the control.

***Java 3D:*** Java 3D is an application programming interface (API) for the Java platform, developed by Sun Microsystems. It is an extension of the java language aimed at the creation and manipulation of 3D scenes.

## IV.3. General architecture of our system:

This section proposes an explanatory diagram which describes the operation of the developed system.

The following figure shows that the user first starts by configuring the parameters of the language games' program and the simulation as input to get started.

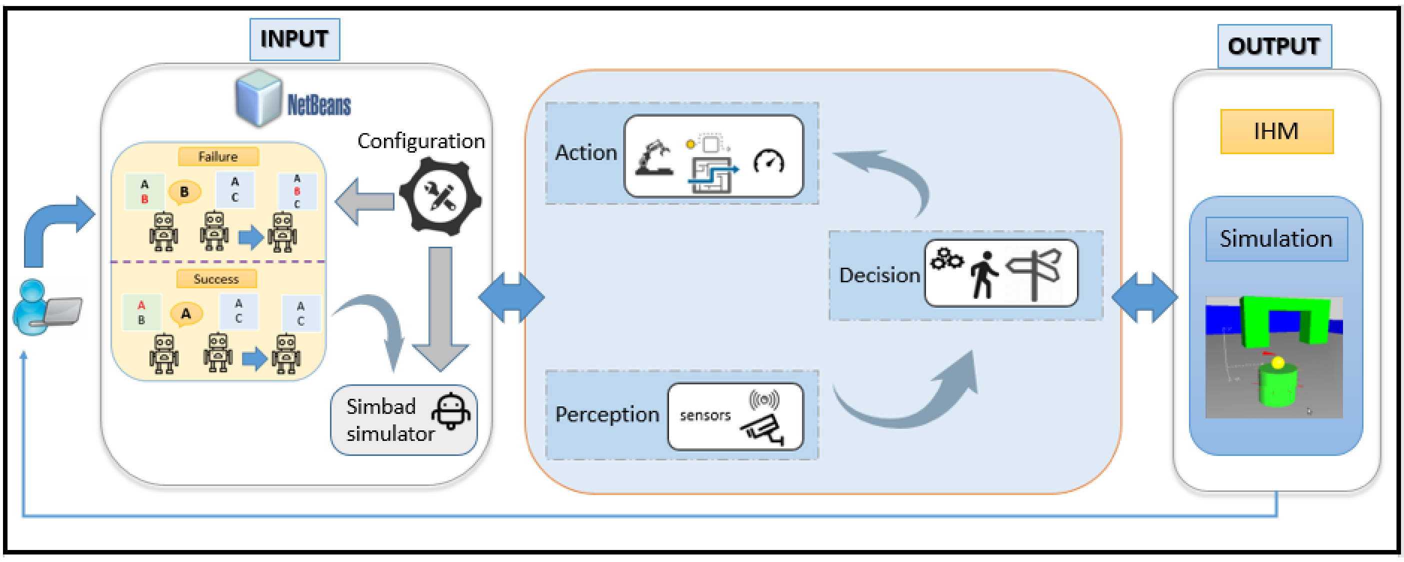


Figure 23: General architecture of our system

## IV.4. Presentation of interfaces:

The figures below represent our system which is composed of several interfaces that would be shown in the following as screenshots.

### IV.4.1. Main interface:

The figure represents the main interface of the application; the user is introduced to two related pathways; he must first press the Language games button to launch its program then go for the simulation.

This interface shows tow prinncipal bouttons :

1) Language Games boutton :lunches the laguage games program on netbeans and leads to get a final global dictionnary.
2) Simulation boutton : lunches the simbad simulation, after that the dictionnary was generated at first place.

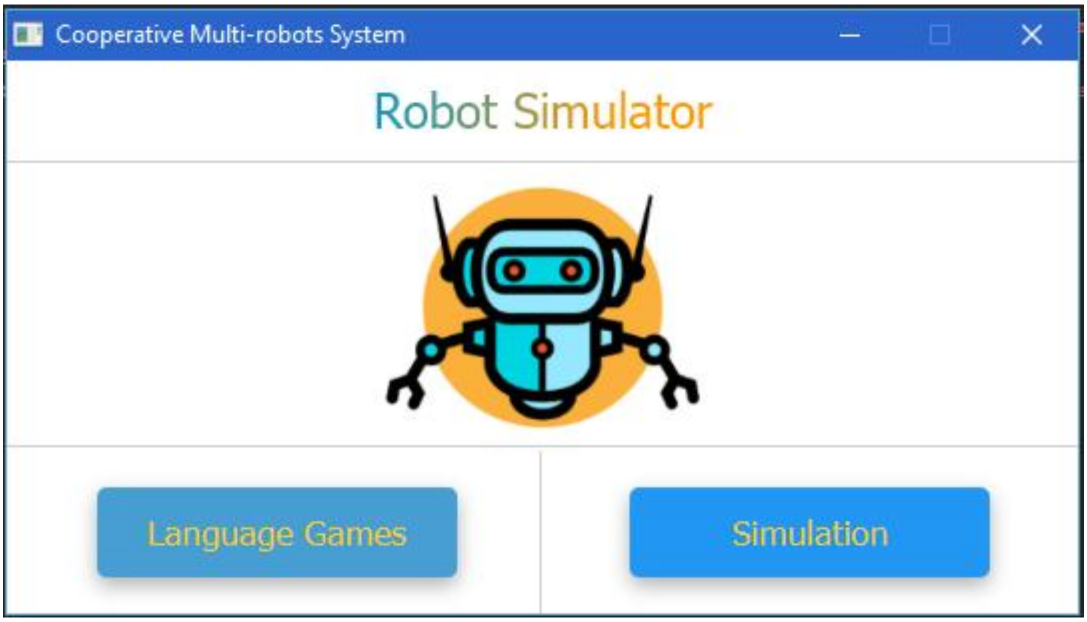


Figure 24: Main interface of the application

### IV.4.2. Language games' interface:

The figure represents the Language games' interface, it allows the user to enter the LG's configuration (the number of robots or agents and the number of cherries),and to run the program with these parameters in clicking the LG's button.

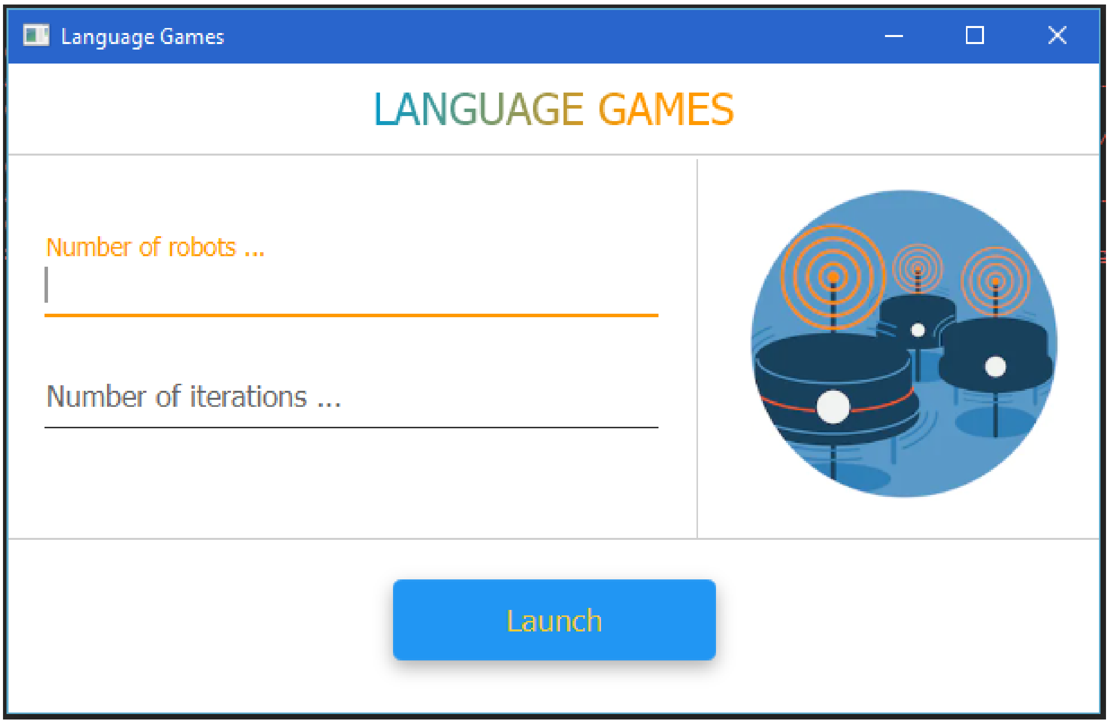
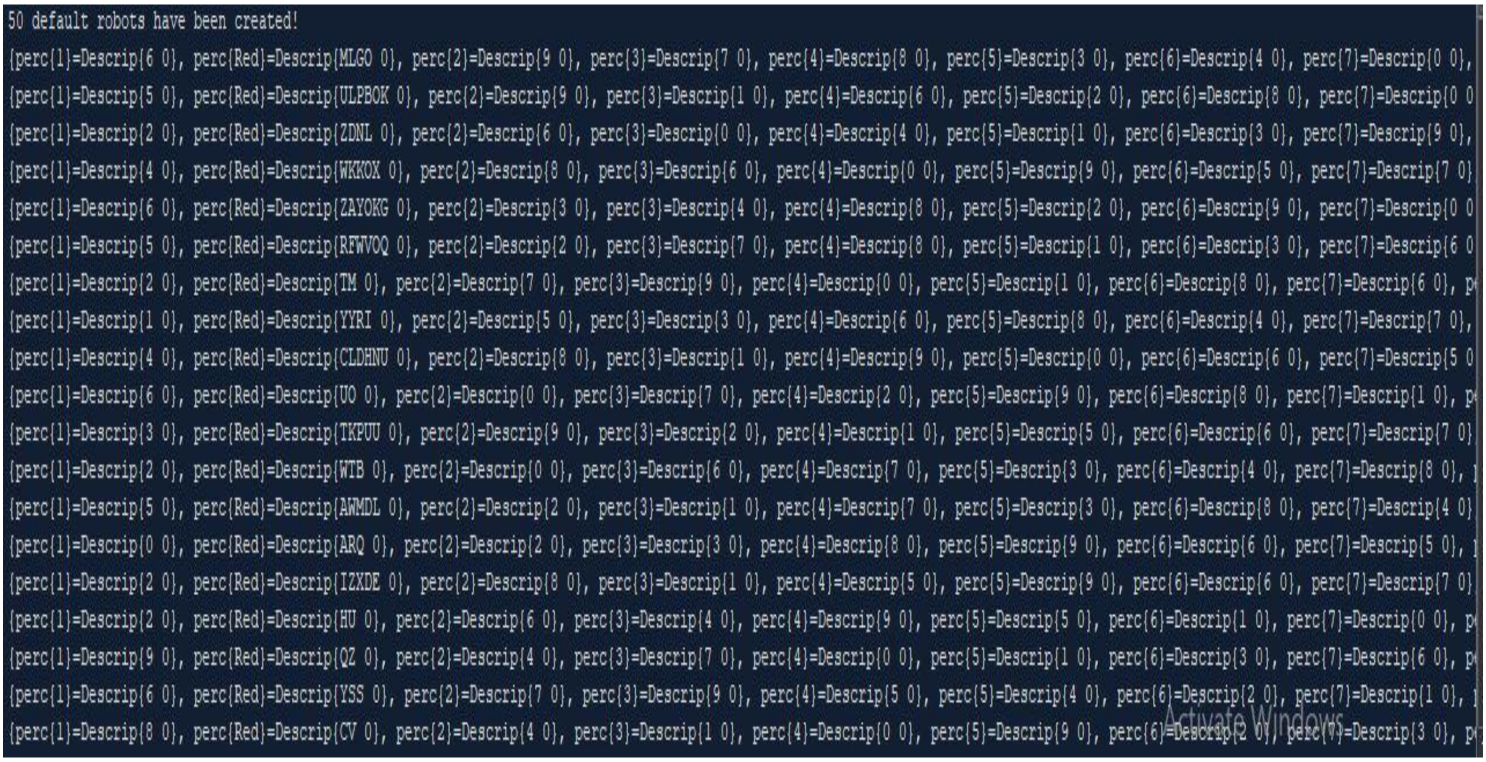


Figure 25: The Language games' interface

The next figures show an example of the output of the LG's program (50 robots and a 10000 iterations) and a display of the final developed lexicon/dictionary which the robots will use later to communicate.

The first figure shows the first 10 robots initial inventory state :



```
50 default robots have been created!
{perc{1}=Descrip{6 0}, perc{Red}=Descrip{MLGO 0}, perc{2}=Descrip{9 0}, perc{3}=Descrip{7 0}, perc{4}=Descrip{8 0}, perc{5}=Descrip{3 0}, perc{6}=Descrip{4 0}, perc{7}=Descrip{0 0},
{perc{1}=Descrip{5 0}, perc{Red}=Descrip{ULPBOK 0}, perc{2}=Descrip{9 0}, perc{3}=Descrip{1 0}, perc{4}=Descrip{6 0}, perc{5}=Descrip{2 0}, perc{6}=Descrip{8 0}, perc{7}=Descrip{0 0
{perc{1}=Descrip{2 0}, perc{Red}=Descrip{ZDNL 0}, perc{2}=Descrip{6 0}, perc{3}=Descrip{0 0}, perc{4}=Descrip{4 0}, perc{5}=Descrip{1 0}, perc{6}=Descrip{3 0}, perc{7}=Descrip{9 0},
{perc{1}=Descrip{4 0}, perc{Red}=Descrip{WKKOX 0}, perc{2}=Descrip{8 0}, perc{3}=Descrip{6 0}, perc{4}=Descrip{0 0}, perc{5}=Descrip{9 0}, perc{6}=Descrip{5 0}, perc{7}=Descrip{7 0}
{perc{1}=Descrip{6 0}, perc{Red}=Descrip{ZAYOKG 0}, perc{2}=Descrip{3 0}, perc{3}=Descrip{4 0}, perc{4}=Descrip{8 0}, perc{5}=Descrip{2 0}, perc{6}=Descrip{9 0}, perc{7}=Descrip{0 0
{perc{1}=Descrip{5 0}, perc{Red}=Descrip{RFWVOQ 0}, perc{2}=Descrip{2 0}, perc{3}=Descrip{7 0}, perc{4}=Descrip{8 0}, perc{5}=Descrip{1 0}, perc{6}=Descrip{3 0}, perc{7}=Descrip{6 0
{perc{1}=Descrip{2 0}, perc{Red}=Descrip{TM 0}, perc{2}=Descrip{7 0}, perc{3}=Descrip{9 0}, perc{4}=Descrip{0 0}, perc{5}=Descrip{1 0}, perc{6}=Descrip{8 0}, perc{7}=Descrip{6 0}, p
{perc{1}=Descrip{1 0}, perc{Red}=Descrip{YYRI 0}, perc{2}=Descrip{5 0}, perc{3}=Descrip{3 0}, perc{4}=Descrip{6 0}, perc{5}=Descrip{8 0}, perc{6}=Descrip{4 0}, perc{7}=Descrip{7 0},
{perc{1}=Descrip{4 0}, perc{Red}=Descrip{CLDHNU 0}, perc{2}=Descrip{8 0}, perc{3}=Descrip{1 0}, perc{4}=Descrip{9 0}, perc{5}=Descrip{0 0}, perc{6}=Descrip{6 0}, perc{7}=Descrip{5 0
{perc{1}=Descrip{6 0}, perc{Red}=Descrip{UO 0}, perc{2}=Descrip{0 0}, perc{3}=Descrip{7 0}, perc{4}=Descrip{2 0}, perc{5}=Descrip{9 0}, perc{6}=Descrip{8 0}, perc{7}=Descrip{1 0}, p
{perc{1}=Descrip{3 0}, perc{Red}=Descrip{TKPUU 0}, perc{2}=Descrip{9 0}, perc{3}=Descrip{2 0}, perc{4}=Descrip{1 0}, perc{5}=Descrip{5 0}, perc{6}=Descrip{6 0}, perc{7}=Descrip{7 0}
{perc{1}=Descrip{2 0}, perc{Red}=Descrip{WTB 0}, perc{2}=Descrip{0 0}, perc{3}=Descrip{6 0}, perc{4}=Descrip{7 0}, perc{5}=Descrip{3 0}, perc{6}=Descrip{4 0}, perc{7}=Descrip{8 0},
{perc{1}=Descrip{5 0}, perc{Red}=Descrip{AWMDL 0}, perc{2}=Descrip{2 0}, perc{3}=Descrip{1 0}, perc{4}=Descrip{7 0}, perc{5}=Descrip{3 0}, perc{6}=Descrip{8 0}, perc{7}=Descrip{4 0}
{perc{1}=Descrip{0 0}, perc{Red}=Descrip{ARQ 0}, perc{2}=Descrip{2 0}, perc{3}=Descrip{3 0}, perc{4}=Descrip{8 0}, perc{5}=Descrip{9 0}, perc{6}=Descrip{6 0}, perc{7}=Descrip{5 0},
{perc{1}=Descrip{2 0}, perc{Red}=Descrip{IZXDE 0}, perc{2}=Descrip{8 0}, perc{3}=Descrip{1 0}, perc{4}=Descrip{5 0}, perc{5}=Descrip{9 0}, perc{6}=Descrip{6 0}, perc{7}=Descrip{7 0}
{perc{1}=Descrip{2 0}, perc{Red}=Descrip{HU 0}, perc{2}=Descrip{6 0}, perc{3}=Descrip{4 0}, perc{4}=Descrip{9 0}, perc{5}=Descrip{5 0}, perc{6}=Descrip{1 0}, perc{7}=Descrip{0 0}, p
{perc{1}=Descrip{9 0}, perc{Red}=Descrip{QZ 0}, perc{2}=Descrip{4 0}, perc{3}=Descrip{7 0}, perc{4}=Descrip{0 0}, perc{5}=Descrip{1 0}, perc{6}=Descrip{3 0}, perc{7}=Descrip{6 0}, p
{perc{1}=Descrip{6 0}, perc{Red}=Descrip{YSS 0}, perc{2}=Descrip{7 0}, perc{3}=Descrip{9 0}, perc{4}=Descrip{5 0}, perc{5}=Descrip{4 0}, perc{6}=Descrip{2 0}, perc{7}=Descrip{1 0},
{perc{1}=Descrip{8 0}, perc{Red}=Descrip{CV 0}, perc{2}=Descrip{4 0}, perc{3}=Descrip{1 0}, perc{4}=Descrip{0 0}, perc{5}=Descrip{9 0}, perc{6}=Descrip{2 0}, perc{7}=Descrip{3 0}, p
```

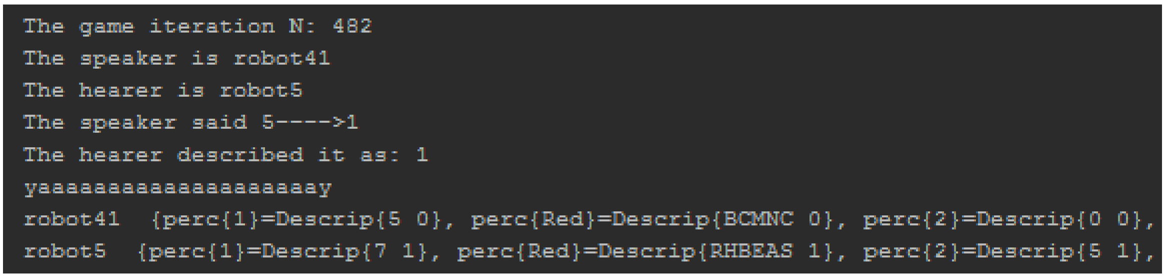
**Figure 26: Example of the initial state of every robots inventory**

We notice that every robot has giving a random discrition for every perception.

Exemple :

1) Robot on line 1 gave the perception ‘Red’ the decription ‘MLGO’.While robot on line 2 gave the perception ‘Red’ the description ‘ULPBCK’, and so did all the other robots .
2) Also we notice that the same robot never gives the same description for two different perceptions and the other robots don’t either.

The second figure shows a success in the game in iteration number 482:



```
The game iteration N: 482
The speaker is robot41
The hearer is robot5
The speaker said 5---->1
The hearer described it as: 1
yaaaaaaaaaaaaaaaaaaay
robot41  {perc{1}=Descrip{5 0}, perc{Red}=Descrip{BCMNC 0}, perc{2}=Descrip{0 0},
robot5  {perc{1}=Descrip{7 1}, perc{Red}=Descrip{RHBEAS 1}, perc{2}=Descrip{5 1},
```

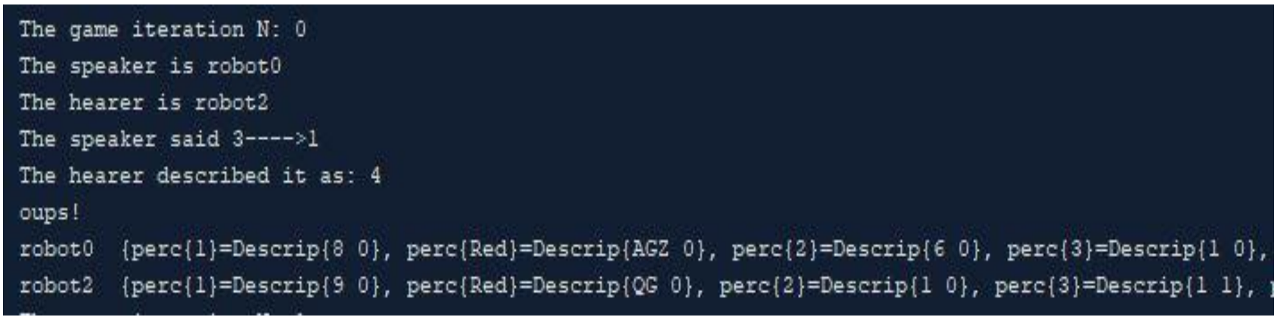
Figure 27: Example of an output of a success

On this iteration, the speaker and hearer robots were choosen randomly :

➔ The speaker robot: is the robot number 41, and the hearer is number 5 .

1. The speaker robot chose the perception ’5’ and described it as ‘1’, after that he sent it to the hearer robot.
2. The hearer checked his inventory and found that he also described it with ‘1’.
3. So the game was a success and both robots updated their inventories and kept ‘1’ as a description for the perception ‘5’.

The third figure shows a failure in the game in iteration number 485:



```
The game iteration N: 0
The speaker is robot0
The hearer is robot2
The speaker said 3---->1
The hearer described it as: 4
oups!
robot0  {perc{1}=Descrip{8 0}, perc{Red}=Descrip{AGZ 0}, perc{2}=Descrip{6 0}, perc{3}=Descrip{1 0},
robot2  {perc{1}=Descrip{9 0}, perc{Red}=Descrip{QG 0}, perc{2}=Descrip{1 0}, perc{3}=Descrip{1 1},
```

Figure 28: Example of an output of a failure

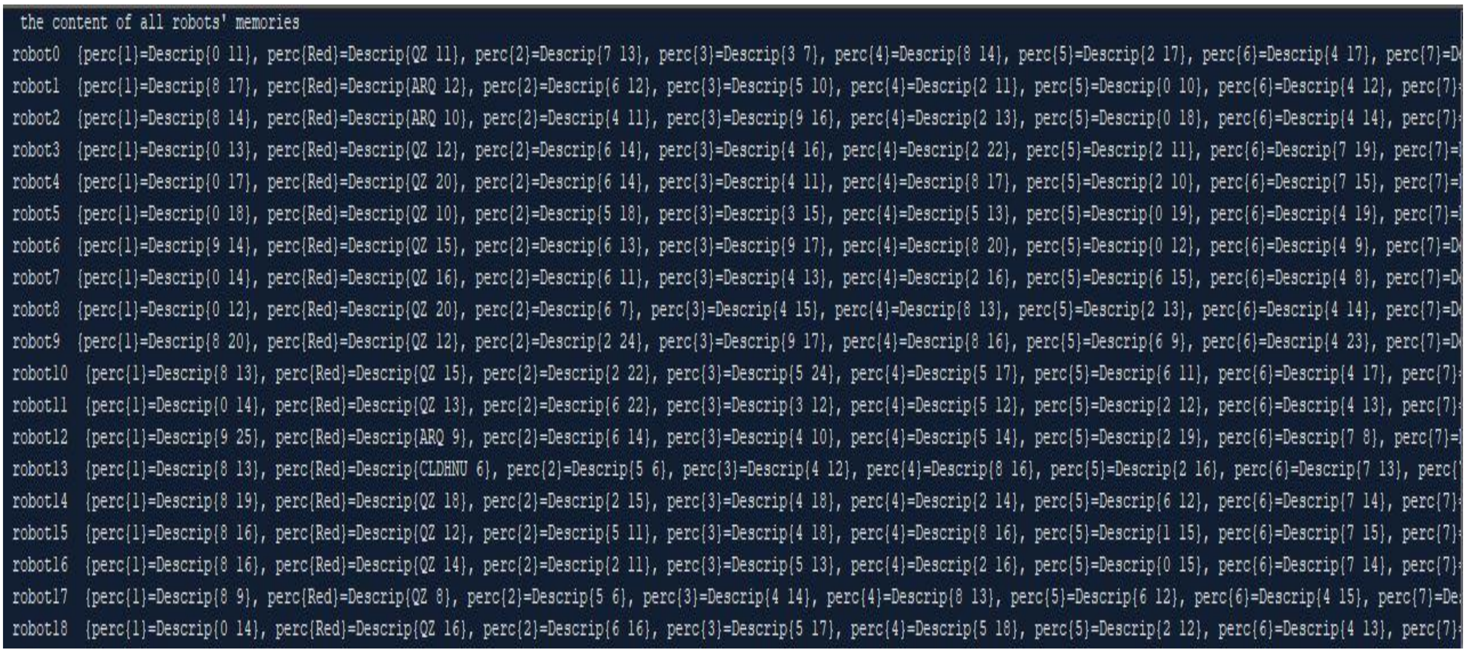
In this iteration the speaker and hearer robots were choosen randomly :

- The speaker is robot number 0
- The hearer is robot number 2

1. The speaker robot chose the perception ‘3’ and generated randomly the description ‘1’ for it then  sent the pair(perception ‘3’,description ‘1’) to the hearer robot .
2. The hearer robot named the perception ‘1’ with the description ‘4’.
3. The game was a failure because it’s not the same description.

➔ And we notice that the hearer robot took the descriptin sent by the speaker and aassociated it to the perception ‘3’ then both robots had updated their inventories.

The fourth figure shows the content of a set of  the robots’ memories after 10000 iterations :

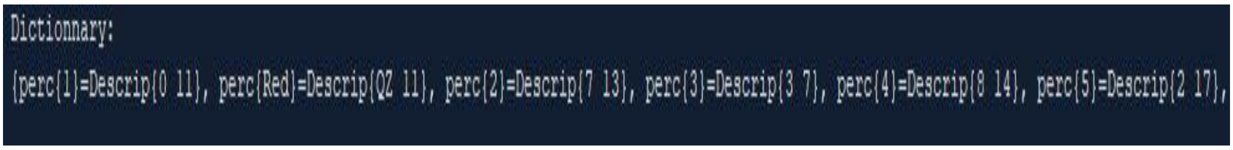


```
 the content of all robots' memories
robot0  {perc{1}=Descrip{0 11}, perc{Red}=Descrip{QZ 11}, perc{2}=Descrip{7 13}, perc{3}=Descrip{3 7}, perc{4}=Descrip{8 14}, perc{5}=Descrip{2 17}, perc{6}=Descrip{4 17}, perc{7}=D
robot1  {perc{1}=Descrip{8 17}, perc{Red}=Descrip{ARQ 12}, perc{2}=Descrip{6 12}, perc{3}=Descrip{5 10}, perc{4}=Descrip{2 11}, perc{5}=Descrip{0 10}, perc{6}=Descrip{4 12}, perc{7}=
robot2  {perc{1}=Descrip{8 14}, perc{Red}=Descrip{ARQ 10}, perc{2}=Descrip{4 11}, perc{3}=Descrip{9 16}, perc{4}=Descrip{2 13}, perc{5}=Descrip{0 18}, perc{6}=Descrip{4 14}, perc{7}=
robot3  {perc{1}=Descrip{0 13}, perc{Red}=Descrip{QZ 12}, perc{2}=Descrip{6 14}, perc{3}=Descrip{4 16}, perc{4}=Descrip{2 22}, perc{5}=Descrip{2 11}, perc{6}=Descrip{7 19}, perc{7}=
robot4  {perc{1}=Descrip{0 17}, perc{Red}=Descrip{QZ 20}, perc{2}=Descrip{6 14}, perc{3}=Descrip{4 11}, perc{4}=Descrip{8 17}, perc{5}=Descrip{2 10}, perc{6}=Descrip{7 15}, perc{7}=
robot5  {perc{1}=Descrip{0 18}, perc{Red}=Descrip{QZ 10}, perc{2}=Descrip{5 18}, perc{3}=Descrip{3 15}, perc{4}=Descrip{5 13}, perc{5}=Descrip{0 19}, perc{6}=Descrip{4 19}, perc{7}=
robot6  {perc{1}=Descrip{9 14}, perc{Red}=Descrip{QZ 15}, perc{2}=Descrip{6 13}, perc{3}=Descrip{9 17}, perc{4}=Descrip{8 20}, perc{5}=Descrip{0 12}, perc{6}=Descrip{4 9}, perc{7}=D
robot7  {perc{1}=Descrip{0 14}, perc{Red}=Descrip{QZ 16}, perc{2}=Descrip{6 11}, perc{3}=Descrip{4 13}, perc{4}=Descrip{2 16}, perc{5}=Descrip{6 15}, perc{6}=Descrip{4 8}, perc{7}=D
robot8  {perc{1}=Descrip{0 12}, perc{Red}=Descrip{QZ 20}, perc{2}=Descrip{6 7}, perc{3}=Descrip{4 15}, perc{4}=Descrip{8 13}, perc{5}=Descrip{2 13}, perc{6}=Descrip{4 14}, perc{7}=D
robot9  {perc{1}=Descrip{8 20}, perc{Red}=Descrip{QZ 12}, perc{2}=Descrip{2 24}, perc{3}=Descrip{9 17}, perc{4}=Descrip{8 16}, perc{5}=Descrip{6 9}, perc{6}=Descrip{4 23}, perc{7}=D
robot10  {perc{1}=Descrip{8 13}, perc{Red}=Descrip{QZ 15}, perc{2}=Descrip{2 22}, perc{3}=Descrip{5 24}, perc{4}=Descrip{5 17}, perc{5}=Descrip{6 11}, perc{6}=Descrip{4 17}, perc{7}=
robot11  {perc{1}=Descrip{0 14}, perc{Red}=Descrip{QZ 13}, perc{2}=Descrip{6 22}, perc{3}=Descrip{3 12}, perc{4}=Descrip{5 12}, perc{5}=Descrip{2 12}, perc{6}=Descrip{4 13}, perc{7}=
robot12  {perc{1}=Descrip{9 25}, perc{Red}=Descrip{ARQ 9}, perc{2}=Descrip{6 14}, perc{3}=Descrip{4 10}, perc{4}=Descrip{5 14}, perc{5}=Descrip{2 19}, perc{6}=Descrip{7 8}, perc{7}=
robot13  {perc{1}=Descrip{8 13}, perc{Red}=Descrip{CLDHNU 6}, perc{2}=Descrip{5 6}, perc{3}=Descrip{4 12}, perc{4}=Descrip{8 16}, perc{5}=Descrip{2 16}, perc{6}=Descrip{7 13}, perc{
robot14  {perc{1}=Descrip{8 19}, perc{Red}=Descrip{QZ 18}, perc{2}=Descrip{2 15}, perc{3}=Descrip{4 18}, perc{4}=Descrip{2 14}, perc{5}=Descrip{6 12}, perc{6}=Descrip{7 14}, perc{7}=
robot15  {perc{1}=Descrip{8 16}, perc{Red}=Descrip{QZ 12}, perc{2}=Descrip{5 11}, perc{3}=Descrip{4 18}, perc{4}=Descrip{8 16}, perc{5}=Descrip{1 15}, perc{6}=Descrip{7 15}, perc{7}=
robot16  {perc{1}=Descrip{8 16}, perc{Red}=Descrip{QZ 14}, perc{2}=Descrip{2 11}, perc{3}=Descrip{5 13}, perc{4}=Descrip{2 16}, perc{5}=Descrip{0 15}, perc{6}=Descrip{7 14}, perc{7}=
robot17  {perc{1}=Descrip{8 9}, perc{Red}=Descrip{QZ 8}, perc{2}=Descrip{5 6}, perc{3}=Descrip{4 14}, perc{4}=Descrip{8 13}, perc{5}=Descrip{6 12}, perc{6}=Descrip{4 15}, perc{7}=De
robot18  {perc{1}=Descrip{0 14}, perc{Red}=Descrip{QZ 16}, perc{2}=Descrip{6 16}, perc{3}=Descrip{5 17}, perc{4}=Descrip{5 18}, perc{5}=Descrip{2 12}, perc{6}=Descrip{4 13}, perc{7}=
```

Figure 29: Example of a final output of the content of robots' memories

➔ We notice that most robots agreed and have the same descriptions for the same perceptions after that they were in a total disagreement.(after a certain time all the community will be in a total agreement and will get to generate a common lexicon).

The sixth figure gives a display of the final developed dictionary:



```
Dictionnary:
{perc{1}=Descrip{0 11}, perc{Red}=Descrip{QZ 11}, perc{2}=Descrip{7 13}, perc{3}=Descrip{3 7}, perc{4}=Descrip{8 14}, perc{5}=Descrip{2 17},
```

Figure 30: Example of an output of a developed lexicon

### IV.4.3. Simulation's interface:

The figure represents the simulation's interface; it is this which allows the user to enter the simulation configuration (the number of locator robots, the number of transporter robot and the number of cherries), to launch the simulation in clicking on the button of simulation and to display the exchange of messages between the robots inside the environment using the developed lexicon from the LG's program.

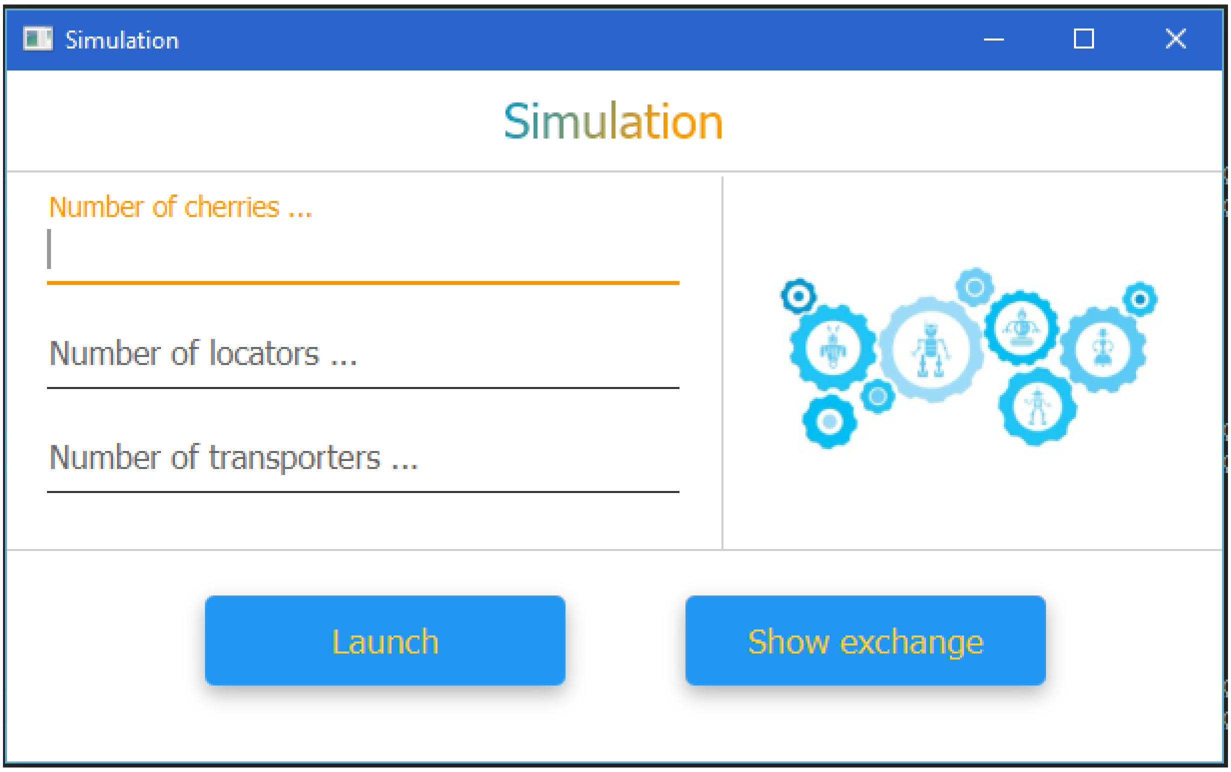


Figure 31: The simulation's interface

***Orientation:***

On this phase from our application we were working on a small community of robots so we took :

1 =>Coordinator robot (that cordinats between locators and transporters and who is respensible of decoding locators messages and specify wich transporter does the work)

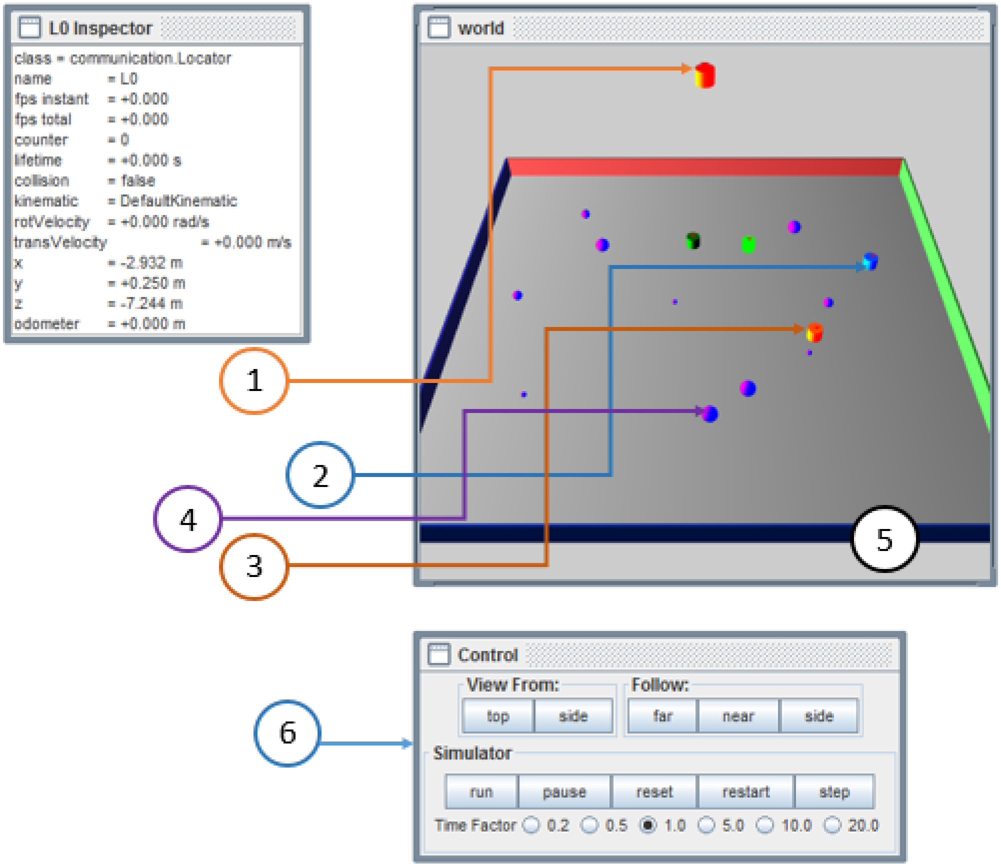
2 => Locator robot (the robot that precieves cherry from the environment,describes the chary,its colo,size and coordonates ).

3 => Transporter robot(it is the one responsible of ditaching cherries from the environment accoding to a specific color ).

4 => Cherry (cherry agent).

5 => Environment’s walls.

6 => Control panel (View form, run, pause, restart…).



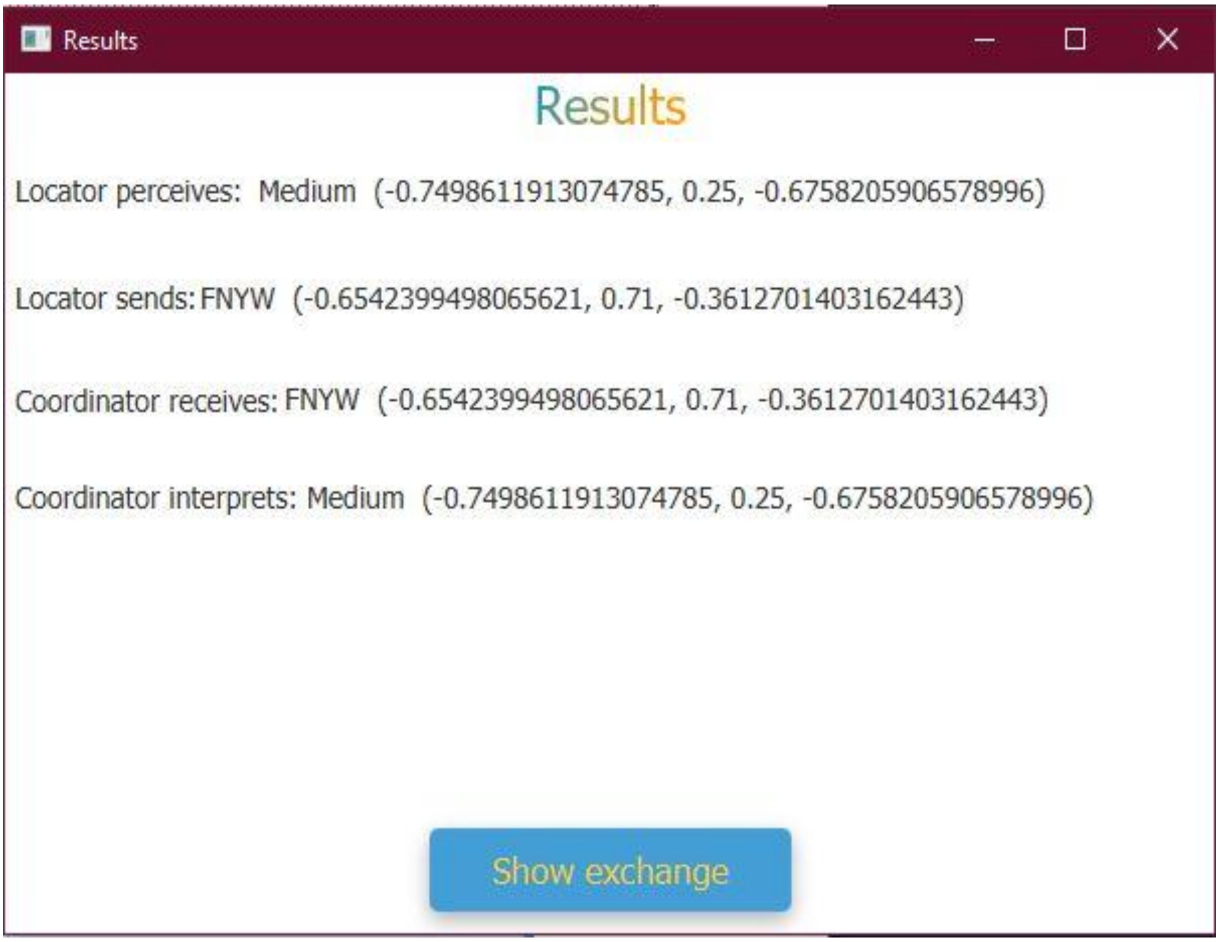
```
class = communication.Locator
name          = L0
fps instant   = +0.000
fps total     = +0.000
counter       = 0
lifetime      = +0.000 s
collision     = false
kinematic     = DefaultKinematic
rotVelocity   = +0.000 rad/s
transVelocity               = +0.000 m/s
x             = -2.932 m
y             = +0.250 m
z             = -7.244 m
odometer      = +0.000 m
```

Time Factor ○ 0.2 ○ 0.5 ● 1.0 ○ 5.0 ○ 10.0 ○ 20.0

*Figure 32: Components of the environment of our system*

## IV.4.4. The messages' exchange interface:

The figure represents the communication interface where the user could make sure that the robots are communicating well and thoroughly.



Results

Locator perceives: Medium (-0.7498611913074785, 0.25, -0.6758205906578996)

Locator sends: FNYW (-0.6542399498065621, 0.71, -0.3612701403162443)

Coordinator receives: FNYW (-0.6542399498065621, 0.71, -0.3612701403162443)

Coordinator interprets: Medium (-0.7498611913074785, 0.25, -0.6758205906578996)

*Figure 33: The communication's interface*

This figure shows the set of messages exchange between the locator and cordinator robots .

We notice that :

-The perceived message by the locator : once the locator finds a cherry while exploring the environment,it gets its size and its position coordinates.

-The sent message by the locator : after getting those pieces of information the locator codes it according to the final generated dictionnary on the langage games phase, and sends it to the coordinator.

-The perceived message by the coordinator: it is the one the locator sent when he founds the cherry.

-The interpreted message by the coordinator: after receiving the perceived message, the coordinator decodes it based on the final dictionnary generated on the langage games phase, in order to know which transporter robot he must send the message to, according to the cherry’s size therfore the proper transporter goes to that specific coordinates and detash that cherry from the environment.

## IV.5. Conclusion:

In this chapter we have presented the environment and technologies that have served us during the development cycle of our application. We also presented the interfaces of the latter, which are simple and easy to use.

# GENERAL CONCLUSION

# GENERAL CONCLUSION:

The objective of this thesis is the simulation of a cooperative multi-robts' system in a dynamic and complex environment using a shared memory developed with the help of language-naming games.

To carry out this work, we had to go through several phases. At first, we did researchon the field of study, this research focused on multi-agents' systems, multi-robots' systems and swarm robotics and their relation to our application. We explained what an agent, a robot and a multi-system are. Then we clarified in explaining transduction problems, deductive reasoning agents, multi-robots' systems and their domain applications. Moreover, we gave a deep explanation on swarm robotics and the evolution and communication in a population of autonomous agents.

Concerning the conception phase, we gave a full explanation on Language games, our usedmulti-robots' system and the navigation. This phase ended byhighlighting our architectural design for the proposed cooperative multi-robots' system withappropriate descriptions for its modules inclosing several explicative organigrams.

In implementation phase, we presented our development environment by indicating the various tools and technologies which are used, such as Netbeans, Scene Builder, Simbad simulator, etc. Then we proved the execution process of our application by depicting theobtained results while presenting some screenshots (interfaces) according to severaluses scenarios.

## REFERENCES:

- [1]: ***The Question Concerning Technology*** (German: *Die Frage nach der Technik*) is a work by Martin Heidegger, in which the author discusses the essence of technology. Heidegger originally published the text in 1954, in *Vorträge und Aufsätze*.

- [2]: Wooldridge, M. and Jennings, N.R. (1995) Intelligent Agents: Theory and Practice.

- Wooldridge, M. An introduction to Multi-Agent systems, 2009

- [3]: FERBER J. Les systemes Multi-Agent- Vers une Intelligence Collective,InterEditions, 1995.

- Ferber J (1999) Multi–agent systems. Addison-Wesley, London

- Ferber J, Muller J (1996) Influences and reaction: a model of situated multiagent systems. In: Proceedings of the 2nd international conference on multiagent systems

- [4]: *Demazeau* Y. From interactions to collective behaviour in agent-based systems,1995 In: Proceedings of the 1st. European Conference on Cognitive Science. Saint-Malo

- [5]: Dennett D, *The Intentional Stance in philosophy of Mind, 1987.*

- [6]: Pattie M, *The agent network architecture (ANA),* ACM SIGART Bulletin, 1991

- Brooks, R. A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation. Volume RA-2, Number 1, 1986.

- [7]: Leslie A, Solving belief problems: toward a task analysis, Cognition, 1998.

- [8]: Grislin, M, Définition d’un cadre pour l’évaluation a priori des Interfaces Homme-Machine dans les systèmes industriels de supervision, 1995.

- [9]: Mandiau,R ,Study of the influence of organizational structure on the efficiency of a multi-agent System, 1999.

- [10]: social humanoid robot developed by Hong Kong-based company Hanson Robotics, 2016.https://builtin.com/robotics

- [11]: Levitt, T.S. and Lawton, D.T. 1990. Qualitative navigation for mobile robots. *Artificial Intelligence Journal*.

- [12]:  Matthias O. Franz , Hanspeter A. Mallot, *Biomimetic robot navigation*, 2000.

- [13]:  Yichuang Jin, ... Alan Winfield, in Neural Systems for Robotics, 1997.

- Kaplan, F, L'émergence d'un lexiquedans une population d'agents autonomes, 2000.

-Dorigo M, şahin E (2004) Swarm robotics – special issue editorial. Auton Robots 17(2–3):111–113

- Dorigo M, Trianni V, şahin E, Groß R, Labella TH, Baldassarre G, Nolfi S, Deneubourg J-L, Mondada F, Floreano D, Gambardella LM (2004) Evolving self-organizing behaviors for a swarm-bot. Auton Robots 17(2–3):223–245

-Nolfi S, Floreano D (2000) Evolutionary robotics: the biology, intelligence, and technology of self-organizing machines. MIT Press/Bradford Books, Cambridge

-Trianni V, Labella TH, Dorigo M (2004) Evolution of direct communication for a swarm-bot performing holeavoidance. In: Dorigo M, Birattari M, Blum C, Gambardella LM, Mondada F, Stützle T (eds) Ant colony optimization and swarm intelligence–proceedings of ANTS 2004– fourth international workshop vol 3172 Lecture notes in computer science. Springer, Berlin Heidelberg New York, pp 131–142

- [14]: Trianni V, Nolfi S, Dorigo M (2006) Cooperative hole avoidance in a swarm-bot. Robot Auton Syst 54(2):97–103

- [15]: Floreano, D, *Evolutionary conditions for the emergence of communication in robots*, 2007.

- [16]: Wittgenstein, L, *Philosophical Investigations*, 1953.

- [17]: De Boer, B. *Imitation games for Complex Utterances*, 2000.

- [18]: Steels, L.*Language games for autonomous robots*, 2001.

- Steels, L. *The Naming Game. The electricity journal* 11(9), 30-33, 2004.

-Baronchelli, A., Felici, M., Loreto, V., Caglioti, E., and Steels, L. (2006). *Sharp transition towards shared vocabularies in multi-agent systems*. **J. Stat. Mech.** 2006:P06014. doi: 10.1088/1742-5468/2006/06/P06014

- [19]:Quine, W. V. O. (2013). *Word and Object*. Cambridge, MA: MIT Press. Rasheed, N., and Amin, S. H. (2016). *Developmental and evolutionary lexicon acquisition in cognitive agents/robots with grounding principle*. **Comput. Intell. Neurosci.** 2016:8571265. doi: 10.1155/2016/8571265.

- [20]: Baronchelli, A. Puglisi, A. *Evolution of Communication and Language in Embodied Agents*, 2008, 2010.

- [21]: Steels, L. *A Self-organizing spatial vocabulary*. Artif. Life 2(3), 319-332, 1995.

- [22]: *Harnad, S. (1990) The Symbol Grounding Problem. Physica D 42: 335-346.*

**-** Baronchelli, A. (2016). *A gentle introduction to the minimal naming game*. **Belgian J. Linguist.** 30, 171–192. doi: 10.1075/bjl.30.08bar

- Ducatelle, F., Di Caro, G. A., Pinciroli, C., and Gambardella, L. M. (2011a). *Self-organized cooperation between robotic swarms*. **Swarm Intell.** 5:73. doi: 10.1007/s11721-011-0053-0

- Floreano, D., Mitri, S., Magnenat, S., and Keller, L. (2007). *Evolutionary conditions for the emergence of communication in robots.Curr. Biol.* 17, 514–519. doi: 10.1016/j.cub.2007.01.058

- Baronchelli, A., Gong, T., Puglisi, A., and Loreto, V. (2010). *Modeling the emergence of universality in color naming patterns*. **Proc. Natl. Acad. Sci. U.S.A.** 107, 2403–2407. doi: 10.1073/pnas.0908533107.

- Nolfi, S., and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines***.** Cambridge, MA: MIT Press.

- [21]: Trianni, V., De Simone, D., Reina, A., and Baronchelli, A. (2016).*Emergence of consensus in a multi-robot network: from abstractmodels to empirical validation*. **IEEE Robot. Autom. Lett.** 1, 348–353.doi: 10.1109/LRA.2016.2519537

- [22]: Miletitch, R., Reina, A., Dorigo, M., and Trianni, V. (2019). *Emergent naming of resources in a foraging robot swarm*. **arXiv:1910.02274.**

- Steels, L. (2003). *Evolving grounded communication for robots*. **Trends Cogn. Sci.** 7, 308–312. doi: 10.1016/S1364-6613(03)00129-3
- Steels, L. (2008). *"The symbol grounding problem has been solved. So what'snext,"* in *Symbols Embodiment: Debates on meaning and cognition*, ed M. Vega(Oxford: Oxford University Press), 223–244.
- Steels, L. (2015). *The Talking Heads Experiment: Origins of Words and Meanings,Vol. 1*. Berlin: Language Science Press.
- Steels, L., and De Beule, J. (2006). "A (very) brief introduction to fluid construction grammar," in *Proceedings of the Third Workshop on Scalable Natural LanguageUnderstanding* (New York, NY: Association for Computational Linguistics),73–80.
- Steels, L., and Loetzsch, M. (2012). The grounded naming game. **Exp. Cult. Lang.Evol.** 3, 41–59. doi: 10.1075/ais.3.04ste

- [23]: Morette, N. *Contribution to mobile robots' navigation: direct model and model based control approach, 2009.*

- [24]: Edraw Max: https://www.edrawsoft.com/edraw-max/

- [25]: Enterprise Architect: https://en.wikipedia.org/wiki/Enterprise_Architect_(software)

- [26]: Java: https://www.freejavaguide.com/history.html

- [27]: NetBeans: https://netbeans.org/about/

- [28]: Scene Builder: https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html

- [29]: Simbad: http://simbad.sourceforge.net/

- [30]: Java 3D: https://en.wikipedia.org/wiki/Java_3D

## *Other webographies:*

- https://www.researchgate.net/
- https://www.geeksforgeeks.org/
- https://scholar.google.com/
- http://gen.lib.rus.ec/
- https://www.sciencedirect.com/