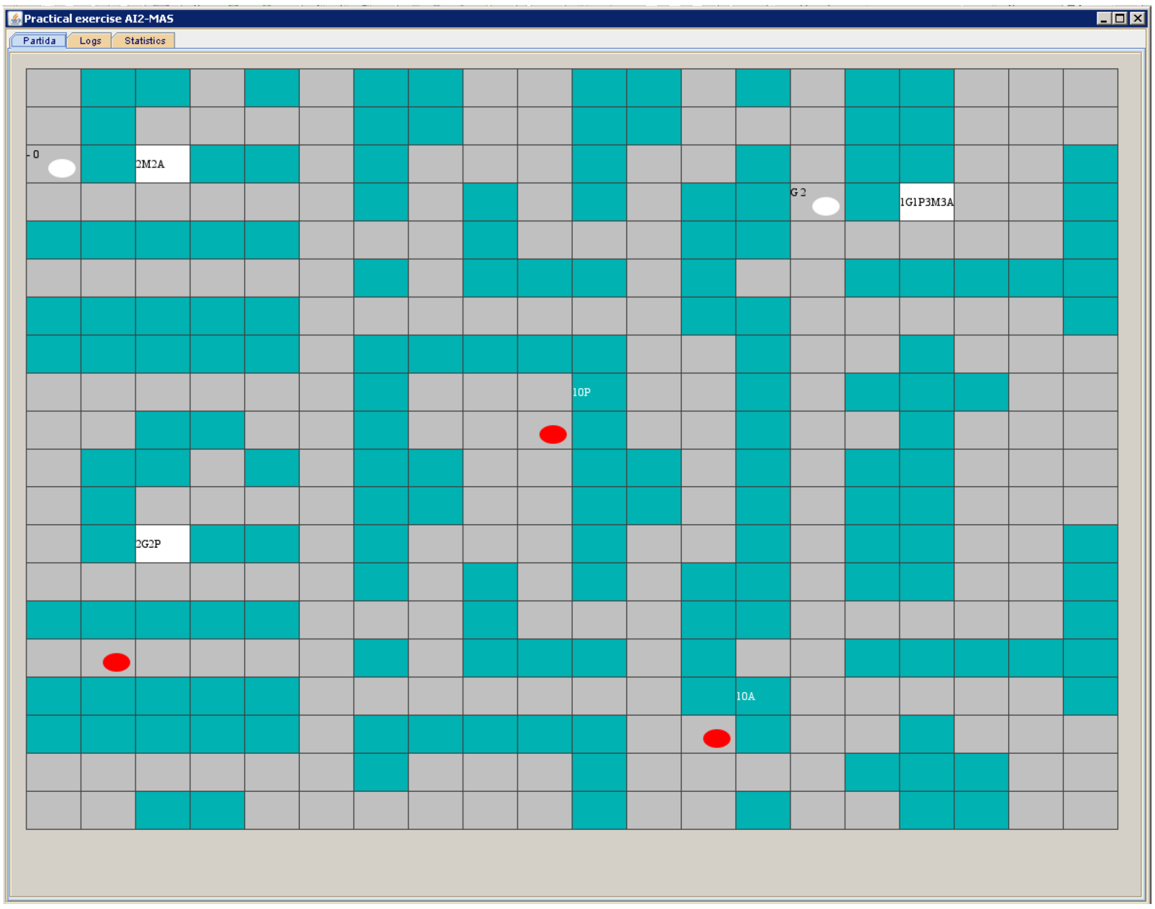
Practical exercise of AI2-MAS-IMAS 2013/2014

Multi-agent systems are a particularly interesting alternative to be applied in distributed systems, in which the different elements act autonomously and they must communicate and coordinate in order to achieve a common task. One of these scenarios is the **efficient cleaning of a city**.

In this problem, there are various types of agents available in the system, endowed with different sensing, acting and communicating abilities, that must cooperate and coordinate in order to recycle as much garbage as possible in a given amount of time. The proposed solution must take into account the characteristics of the map, the different types of garbage and the number of points given when this garbage is recycled.

Elements of the city



This practical exercise aims to simulate the cleaning of a city by different agents which must act in an intelligent and coordinated way. These agents try to harvest the garbage as soon as possible and try to maximize the number of points given for the recycling of this garbage. The agents involved in this problem will be of two kinds: scouts and harvesters, both having different abilities. We will consider the city divided into a set of grid cells. Within this area there are *buildings* (which may contain *garbage*), *streets* where scouts and harvesters may be located, and *recycling centers*. We will add the constraint that *in a certain street cell there can only be a unique vehicle (scout or* *harvester) in each moment*. The previous figure shows a possible configuration during the execution of the system, where there are some grey cells representing streets, some cyan cells representing buildings and three white cells representing recycling centers. Some buildings may contain garbage and this is represented with a string. For example, “10P” means that the building has ten units of plastic garbage (G=Glass, P=Plastic, M=Metal, A=Paper). Similar strings are used to indicate the number of points given for each unit of garbage in the recycling centers. For example, one of the recycling centers has “2M2A” so it gives 2 points for each unit of metal or paper. In the figure, there are also three scouts (red ellipses) and two harvesters (white ellipses). The two characters next to each harvester represent the current type of garbage and the number of units it is carrying. In this example, one harvester has “G2” meaning that it is carrying 2 units of glass, and the other harvester has “-0” meaning that it is not currently carrying garbage.

Mechanisms and interactions

There are three basic kinds of actions that have to be done: **garbage detecting, garbage harvesting** and **garbage recycling**. A certain building may contain some units of garbage. The number of units of garbage for each building is set at the beginning of the game but new units of garbage may appear at any moment during the game in any of the buildings. This garbage may be of different kinds: *plastic*, *metal*, *glass* or *paper*. A concrete building only contains one kind of garbage. For example, it could contain 5 units of metal garbage. This garbage must be detected by exploring the map. Once detected, it must be harvested and then disposed in one of the available recycling centers. Each recycling center only deals with some of the kinds of garbage. The recycling centers give a number of points for each unit of garbage disposed. For example, there could be a recycling center that gives 2 points for each unit of plastic garbage and 3 points for each unit of paper. This recycling center does not deal with metal and glass garbage. For each kind of garbage, there will be at least one recycling center in the city that accepts it.

The shape of the city will be known a priori whereas the location of the garbage will be unknown. Only the scouts will be able to detect new garbage. Each scout will have a visual range limited to the cell where it is located plus the 8 cells surrounding it. As soon as the visual range of a scout contains a building with garbage, this garbage will be discovered. Notice that the previous figure shows a configuration where we observe two buildings with garbage. This is because they are located in one of the cells surrounding one of the scouts. Since this moment, this garbage will be visible. Probably there are more buildings with garbage but they have not been discovered yet. So, the scouts will have to move through the street cells in order to **detect garbage**. Once a new cell with garbage is discovered, any of the harvesters will be able to pick it and take it to a recycling center. The positions of the recycling centers and the number of points they give for each kind of garbage will be known by the scouts and harvesters since the beginning of the game.

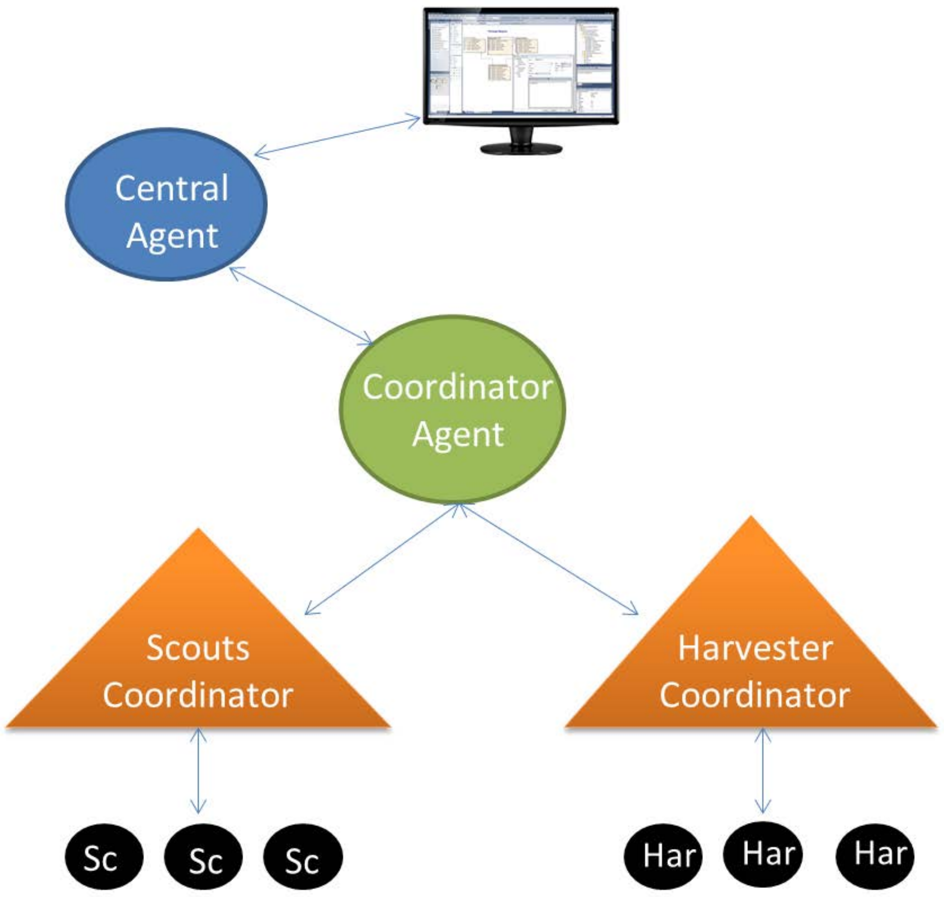
Each scout and harvester will have its corresponding agent (**scout_agent** and **harvester_agent**, respectively). These agents keep their current position and their internal state (for example, a harvester keeps the number of garbage units it is carrying and their kind) and make their own decisions or wait for orders from their managers (depending on the design of the multi-agent system). Scouts and harvesters have different tasks to perform:

- *Scouts*: They explore the map. These vehicles will move through the city discovering the buildings that have garbage. They cannot harvest garbage. They can move, horizontally or vertically, 1 cell per turn.
- *Harvesters*: They harvest garbage and bring it to a recycling center. They can move, horizontally or vertically, 1 cell per turn. In order to harvest garbage, they must be situated in a cell adjacent to the building containing garbage (horizontally, vertically or diagonally) and remain there for some time (1 turn per garbage unit). Each harvester can harvest one or more kinds of garbage but it can only carry one kind of garbage at the same time. Moreover, each harvester will have a maximum number of units of garbage that it can carry (the same for each type of garbage). When they have harvested garbage, they can go to harvest in another building if the maximum number of units has not been reached or they can go to recycle this garbage. To do this, a harvester has to be situated in a cell adjacent to a recycling center (horizontally, vertically or diagonally) that allows the kind of garbage it is carrying and stay there for some time (1 turn per garbage unit). Several harvesters can be harvesting garbage from the same building or disposing garbage in the same recycling center at the same time.

The agents representing the scouts and the harvesters can be coordinated using several methods: they may have an additional agent acting as a manager or as coordinator (e.g., scout_coordinator), they may have several coordinators of different levels (e.g., harvester_coordinator, metal_harvester_coordinator, glass_harvester_coordinator, etc.), there can be coordinators for different parts of the city, they may also interact with each other using no coordinators, etc. For example, if we use coordinators, each one of them may have a list with the buildings containing garbage at the moment and use it to make decisions. For example, harvest_coordinator, knowing that there is a limited number of harvesters, must determine how to assign the different harvesters to each one of the buildings with garbage, depending on the kind and amount of garbage, the capacity of the harvesters and their location in the map. Several strategies can be followed, like assigning 1 harvester/building, sending some harvesters to the same building in order to harvest its garbage earlier, sending a harvester to a recycling center which gives less points than another one but it is very much closer, decide routes for the scouts that do not interfere in the harvesting tasks, etc.

In any implementation of this simulation, there has to be a **coordinator_agent** which centralizes the orders to be executed in each turn. This coordinator_agent knows the changes that dynamically happen in the city (e.g., movement of the vehicles, new garbage detection, garbage harvested, etc.). In order to simulate and control what is happening, there is also a **central_agent** which executes orders to update the state of the world. This agent is the one that, in a nutshell, keeps the state of the city and shows it using a graphical interface.

The next figure depicts a part of the agent architecture that must be implemented:



The communication between the Coordinator_Agent and the scouts and harvesters will depend on the architecture chosen for the multi-agent system.

Information flow

As it can be observed in the previous figure, there are different agents involved in the cleaning of the city. In order to show more clearly how the system works and which role is played by each actor, we describe the basic steps of the functioning:

- Initially, the Central_Agent loads the configuration of the city from a file. This configuration contains information about the probability to generate new garbage, the number of turns, the time to think (in milliseconds), the number of cells, and the type of each one. In the case of buildings, the number and type of garbage units in the initial state of the simulation. In the case of recycling centers, the kinds of garbage and the number of points given for each kind. It also loads the number of scouts and harvesters. For each harvester, the file also contains the kinds of garbage it can harvest and its capacity.

- The Central_Agent randomly places all the scouts and harvesters in the city. The main parameters of the game, the shape of the map, the positions of the vehicles, the list of buildings with garbage detected (if there are any), the positions of the recycling centers and the number of points they give for each kind of garbage are sent to the Coordinator_Agent. The different Scout_Agents and Harvester_Agents are created and their corresponding information is sent to them.

- The agents (harvesters and scouts) send an Ok message to their respective coordinator saying that everything is correct and that the simulation is ready to start. Then, this information is sent up to the Central_Agent following the coordination flow of the agents. The simulation is ready to start.

- Each Scout_Agent, Harvester_Agent, or their specific coordinator depending on the architecture used will "think" the strategy to follow. This strategy causes that they want to change the position of some elements or make some actions (harvest or recycle). This desire is transmitted to the Coordinator_Agent, which receives all the proposals and transmits them to the Central_Agent. The Central_Agent, in concrete periods of time (turns) waits for the Coordinator_Agent to send a list with all these movements or actions of scouts and harvesters. If there is none, the list is empty.

- Once the Central_Agent receives a list of movements and actions, it performs them (as long as it is possible – e.g., a movement can not be done if it goes to a cell with another vehicle; if two vehicles want to go to the same position at the same time, none of them will move; two vehicles can not interchange their positions in the same turn) and informs the Coordinator_Agent of the result of the actions and provides a list of new buildings with garbage detected (if there are any). The Central_Agent also decides to create new garbage with a certain probability. In this case, a new unit of a random type of garbage will be placed in a random building.

- After having provided the list of movements and actions to perform, coordinators must check that these actions have actually been carried out, and update their information. If they have not been performed, maybe they have to re-plan the situation (e.g., change the assignment of harvester to a building, or change the route leading to a recycling center).

This process is basically divided into two stages: initialization and simulation. The *initialization* is used to place all the agents in their corresponding position and initialize the knowledge they must keep. The *simulation*, which works using synchronous turns, is used to keep changing the positions and states of the agents.

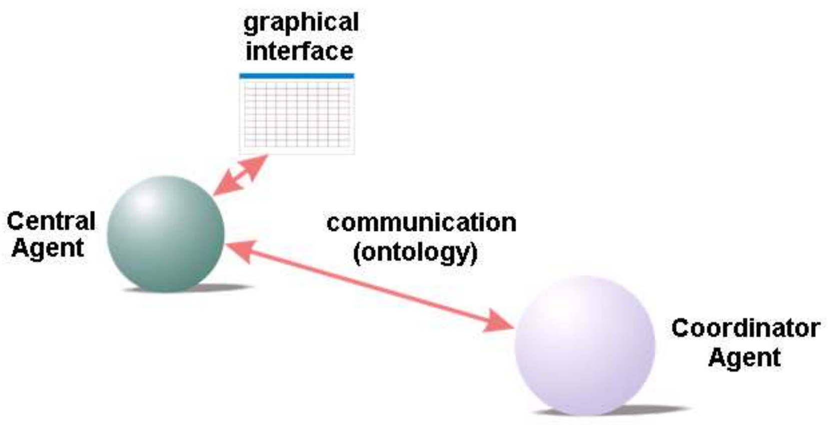
The Central_Agent will keep some statistics and a log of the executed tasks in order to evaluate the work of the different actors. Once the number of turns specified in the configuration file is reached, the simulation finishes and the statistics are shown. Basically, the statistics that will be used to evaluate the performance of the agents are:

- Percentage of the number of points obtained over the maximum number of points for this map
- Percentage of units of garbage harvested
- Percentage of buildings with garbage discovered

Practical and theoretical classes

During the course, different topics related to the implementation of a multi-agent system (MAS) will be explained, both in practical and theoretical classes. It is mandatory to represent the knowledge transmitted among the agents, to decide the communication, coordination and negotiation protocols, to decide the synchronization process of all the information flows, etc. All this knowledge must be applied during the implementation of each team, and all the decisions made will have to be justified. This is one of the important issues to explain in the e-portfolio of each team and it will be highly evaluated.

In order to make the learning of the most basic aspects of MAS programming with JADE easier, you will be given a simple template for the Central_Agent, which interacts with a graphical interface and a (basic) Coordinator_Agent. In the next figure, the different interactions are depicted:



With this base, you will be able to: compile a MAS with the JADE library, run agents, observe how they transmit messages to each other (sending and receiving), their format, etc. Moreover, in the library there is an excellent book to start to study JADE programming (Developing Multi-Agent Systems with JADE).

Tests

An exhaustive set of tests must be performed. Some of the parameters to experiment with are:

- Number of scouts and harvesters.
- City topology and size
- Probability of creation of new garbage
- Number of recycling centers
- …