Universitat Rovira i Virgili

Introduction to MultiAgent Systems

Activity 1

First analysis of the practical exercise: environment, agent architecture and agent properties

Authors:
Hugo Bertiche Argila
Axel Brando Guillaumes
Julià Camps Sereix
María Leyva Vallina
Vicent Roig Ripoll

Supervisor:
Dr. Jordi Pujol

November 21, 2015

Contents

Introduction

Goal

In this report we want to make a first approach of what will conform the main structure of our MAS. We will discuss main issues related to the environment and agents, their architecture and properties.

Description

In this activity, we will study the theoretical concepts described in the first lectures and relate them to the practical exercise of the MAS course.

First, we will analyse the characteristics of the environment that our practical exercise presents. We will evaluate each possible feature and decide how will our environment be:

- Accessible or inaccessible.
- Deterministic or non-deterministic.
- Episodic or non-episodic.
- Static or dynamic.
- Discrete or continuous.

After this, we will proceed to analyse our agents and their architecture, to determine if every single one of them is reactive, deliberative or hybrid and why.

Finally, we will study the different properties our agents can have, and explain which ones will they have, giving again a reasoned explanation.

Characteristics of the environment

In this section the characteristics of the given problem environment are described and justified.

The environment is the scenario where the agents-system will be interacting. Therefore, an accurate description of this environment will provide of the capabilities that the agents might need in order to accomplish their objectives.

Although, correct description of the environment will not necessarily lead to a good design of the agents-system, an incorrect description of it, will indeed, lead to a wrong design.

The characteristics that are being considered in order to describe our agents-system environment can be seen as binary discrete attributes (binary in most cases).

Here we list the used characteristics, with their values for each one, and the corresponding justification, for the value assigned, among the current problem description:

Accessible or Inaccessible

In the problem description it is specified that agents on the system might be warned of changes on it on the same *step*[1] that they take place. From this we consider it to be an **Accessible Environment**, because the data model of the environment is available for the agents.

Deterministic or Non-deterministic

Our environment is a **Deterministic Environment**, because there is always only one single correctly predictable outcome to any given action on a given state of the environment.

Episodic or Non-Episodic

Environment is a **Non-Episodic Environment**, as the decisions of the agents in the system needs to consider future states of the environment, in order to optimise the performance of the system resources. (Example: sending a rural agent through an avalanche current path, but knowing that this current avalanche will have finished when the agent reaches it).

Static or Dynamic

For taking this decision we have to take into account the special characteristics of our *SystemAgent*. This "agent" it is in charged of the control of all arbitrary changes in the environment, so it behaves as a simulation of a more "real" environment. This means, *SystemAgent* will be the one who will make all the changes in the environment. If we consider the *SystemAgent* an "agent" of our agents-system, the environment is a **Static Environment**. However, if we do not consider *SystemAgent* a "agent" of the agents-system "agents", and just as the process en-charged to control the environment simulation, and we encapsulate it in a black-box, which provides a data model of the changes of the environment at each *step*, in this case the environment is a **Dynamic Environment**.

Discrete or Continuous

Environment is a **Discrete Environment**, as there is a finite number of possible actions to be performed on it at a current state.

[1] We will understand a *step* as the minimum time unit in the practical exercise system.

Agent architecture

We have to determine the level of sophistication and rationality of the agents in our MAS. We have to choose if they will be reactive, deliberative or hybrid. Reactive agents do not make any kind of plan. They only execute pre-set behaviours. On the contrary, deliberative agents do make plans: they maintain an internal state, try to predict the effect of their actions and choose their behaviour basing on this.

Although the line between reactive and deliberative agents can be somewhat blurry. In our implementation all types of agents (reactive or deliberative) have an internal state. We consider that the difference between both are if they do or not the main calculation about his actions. Here we describe one system at each extreme as well as two others that mix reactive and deliberative reasoning.

System Agent

This agent might not be considered neither as deliberative, reactive nor hybrid, since it controls all the environment, but does not really have the behaviour of an agent.

Agent Coordinator

Is deliberative, as it has to take into account environment state and plan its actions. According to injured mountaineers and avalanches positions, it has to decide whether it is rural agents or helicopter agents responsibility to pick up each mountaineer.

Rural Agent Coordinator

Has a deliberative architecture. It needs to know mountaineers and avalanches positions as well as paths, in order to decide which rural agent to send to which injured mountaineer.

Helicopter Coordinator

Has a deliberative architecture. As Rural Agent Coordinator, needs to know mountaineers positions, in order to calculate which helicopter to send to perform the rescue.

Rural Agent

Has a reactive architecture. This agent only needs to know which path to follow, and how to quickly react to avalanches. In special cases, it just asks the coordinator what to do next.

Helicopter Agent

Has a reactive architecture. Given an objective mountaineer, advances in its direction. Once is there, picks up mountaineer/s. If after rescuing its not at full capacity it asks the coordinator what to do next, go rescue another one or come back to hospital.

Agent Properties

We analysed the properties on each type of agents, based on the agents architecture and the description of the environment characteristics, described in the previous sections, that the agents should incorporate in order to be able to accomplish their purposes in this practical exercise.

Possible Properties Description

The properties that we consider that an agent can have are the following:

Reactivity

We consider that an agent is "reactive", if it is capable to maintain an ongoing interaction with its environment, and responds to changes that occur in it (in time for the response to be useful).

Pro-activeness

We consider that an agent is "proactive", if it is capable to act generate new goals taking the initiative and anticipating to needs and problems. (Example: having a rural agent going out to explore a zones that are more likely to become isolated).

Social Ability

We consider that an agent is "proactive", if it is capable to interact, and therefore communicate, with other agents via some kind of agent-communication language.

Flexibility

We consider that an agent is "flexible", if it is capable to perform *flexible* actions. By *flexible* we mean: Reactive, Proactive and Social.

Autonomy

We consider that an agent is "autonomous", if it is capable to pursue goals in an autonomous way, without direct continuous interaction/commands from the user, the agent must determine the best way to achieve a given goal.

Rationality

We consider that an agent is "rational", if it is capable to distinguish which actions lead achieve its goals from those that would prevent its goals from being achieved, and will try to perform only actions in order to achieve them.

Reasoning capabilities

We consider that an agent is "reasoning capable", if it is capable to reason. This property is an essential aspect for: implementing intelligent/rational behaviours, having the ability to infer and extrapolate based on current knowledge, having the capacity of planning. We could say that this is the characteristic that distinguishes an intelligent agent from a more "robotic" *reactive-like* agent.

Learning

We consider that an agent is "learning capable", if it is capable to improve its performance from *experiences*[2].

Temporal continuity

We consider that an agent is "temporal continuous", if his life cycle is not linear, it continues "alive"[3] until it's "killed"[4].

Mobility

We consider that an agent is "mobile", if it is capable to move physically through a network to different platforms during his life cycle.

Properties of the Agents

From the previous possible properties that we consider an agent can have. We classify the agents that will appear in our agents-system according to which ones fulfil each property:

Reactive Agents

Rural-agent and *Helicopter* agent, but although they are capable to perform "reactive" actions. (For example: if an helicopter has to land on a hospital but the helicopter platform is occupied, he must wait until it is free and adapt to this unexpected situation.)

Proactive Agents

On our MAS[5] approach, we specified both agents behaviour to wait for their coordinators to send them a request to perform a specific task and so our agents lack of pro-activeness, this has been decided in order to avoid waisting resources of the system (in the problem description it is specified that agents should not be exploring the area on their one, except when an injured mountaineer is there, and that the use of resources should be optimised).

Social Agents

We can affirm that all our agents are **social**, since all of them are constantly interacting with other agents. *SystemAgent* comunicates with *AgentCoordinator* to send it information about new injured people appearing in the map, and this one communicates with *RuralAgentCoordinator* and *HelicopterCordinator* to send them this same information, and to receive from them their best candidate to perform the rescues. These two coordinators also interact with their corespondent agents (*RuralAgents* and *HelicopterAgents*) in order to transmit them orders.

[2] We will understand *experiences* as *data* to which the agent has access to, and from which he can extract conclusions.

[3] We say that a process is "alive" while it's running, and that it's "dead", when it has finished running.

[4] To "kill" a process is the action to make it finish (to terminate it).

[5] Multi-Agent-System

Flexible Agents

In order to be flexible, an agent must satisfy all this three properties: **Re-activeness**, **Pro-activeness** and **Sociability**. The closes agents to be *flexible* are the *Rural-agent* and *Helicopter* agent, but although they are capable to perform "reactive" actions, and can communicate and interact with other agents (so they can perform "social" actions), they are not able to perform "proactive" actions, and consequently, they can not be considered *flexible* agents.

Autonomous Agents

We consider that *SystemAgent* posses **autonomy**, since it does all its functions without any external interaction, and **temporal continuity**, because it is constantly running, updating the map with new injured people, avalanches and sending this new information to the *AgentCoordinator*. In the other hand, we cannot state that *AgentCoordinator* have temporal continuity, since it only reacts when it receives new information from *SystemAgent*, although, it can take it is own decisions,In this case, deciding which agent (*RuralAgent* or *HelicopterAgent*) should be sent to perform a rescue, so it is **autonomous**.

We can also consider *RuralAgentCoordinator* and *HelicopterCordinator* **autonomous, but with some restrictions**. They can take their own decisions (which one is the best of their agents to send to a rescue, and which is the best route for it), but they cannot send it directly until *AgentCoordinator* confirms them so.

Rational Agents

We can also say that all our Coordinators are **rational**, seeing that all of them work to get their goal (the least cost) and will not work in any case to prevent it to be achieved.

Reasoning Capable Agents

Still, we can regard these same agents as possessors of **reasoning capabilities**, after all, they have the capacity to make plans (calculate routes for *RuralAgents* and *HelicopterAgents*), for what they must have knowledge of the environment and be able to apply it to solve their problem (which agent should I send to attend this injured mountaineer?).

Learning Capable Agents

At the moment, any of our agents posses the **learning** property, but we could consider to add them a learning capacity in favour of improving the performance of the whole system.

Temporal Continuous Agents

We can say that all agents in our system posses the **temporal continuity** property.

Mobile Agents

Finally, all our agents lack the property of **mobility**: they will not be able to move physically to another machine.