UNIVERSITAT ROVIRA i VIRGILI

Introduction to **M**ulti **A**gents **S**ystems

Activity 1

Course 2020-2021
October 21st, 2020

Students:

Roger Marrugat-	-	*roger.marrugat@estudiantat.upc.edu*
Albert Ferrando	-	*albert.ferrando.garrido@estudiantat.upc.edu*
Victor Badenas	-	*victor.badenas.crespo@estudiantat.upc.edu*
Ronald Rivera	-	*ronald.rivera@estudiantat.upc.edu*
Laia Seijas	-	*laia.seijas@estudiantat.upc.edu*

Abstract

The aim of this activity is to apply the knowledge acquired in theoretical classes to a practical work, in which the characteristics of the environment, the agents architecture and the agent properties must be defined.

Contents

1. Overview

The goal of the Practical of the *Introduction to Multi Agents Systems* practical course is to create a Fuzzy Agent-based Decision Support System (FA-DSS), which will perform two main actions:

1. Prepare or Initialize the Fuzzy Agent-based System (according to the requirements sent by the user).
2. Handle the requests from users, evaluating the inputs using a Fuzzy Inference Engine.

The system should be open, scalable, able to deal or be used under different circumstances and take into account a variety of strategies and topologies.

In this first Activity the goal will be to define the system's environment and its agents.

2. System architecture

The aforementioned described system includes three different agents: *User Agent*, *Manager Agent* and *Fuzzy Agent*. Figure 1 shows a high-level architecture and the interactions between the agents.

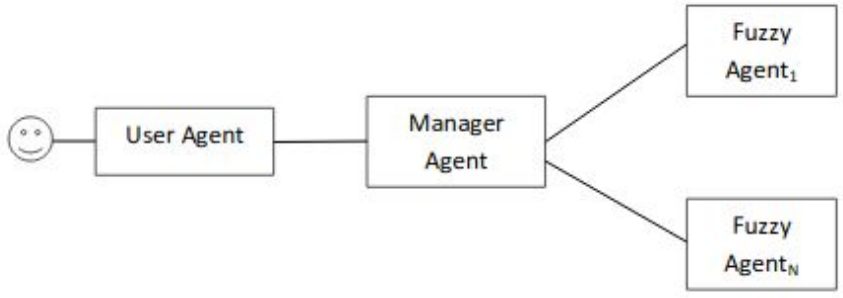

Figure 1. Basic schema of system architecture

- ***User Agent*:** Interacts with the user in order to start the MAS (Multi Agent System) according to the requirements that the user specifies and provides the user an interface to send requests to the system and receive an appropriate response.

- ***Manager Agent:*** Manages the requests received from the *User Agent* and creates *Fuzzy Agents* to handle them. Acts as the gateway between the *User Agent* and the set of *Fuzzy Agents*.

- ***Fuzzy Agent:*** Defined by a *Fuzzy Set*, they embed a *Fuzzy Inference Engine* and represent the set of available resources which are used to handle the data and infer the final response that solves the problem.

3. Environment characteristics

3.1. Accessible vs. Inaccessible environment

An accessible environment is one in which the agent can obtain complete, accurate and up-to-date information about the environment's state.

Proceeding from the assumption that the user agent will provide all the necessary inputs and these inputs will be precise and correct, we can determine that our system will present an accessible environment.
In addition, we have taken into consideration additional scenarios that would result in an inaccessible model, the scenarios are as follows:

- Missing input data
- Incorrect inputs provided by the user agent
- Fuzzy Inputs
- Variability in data inputs

3.2. Deterministic vs. Non-deterministic environment

A deterministic environment is one in which any action has a single guaranteed effect. There is no uncertainty about the state that will result from performing an action.

Starting from the assumption that the system produces a unique consistent output given it is provided with consistent entry values, we establish that it is a deterministic environment.

We have defined multiple factors that would cause this system become non-deterministic:

- Random factor present in the output
- Output could not be guaranteed
- Static or define "a priori" output

3.3. Episodic vs. Non-episodic environment

If the environment is episodic, the results obtained will be influenced from previous experiences.
Based on the reviewed information we have determined that our system follows episodic behavior due it should not consider previous executions.
The presence of the following environments would cause that the system follow a non-episodic behavior:

- Independent inputs values between each other
- A component of machine learning present in the system

3.4. Static vs. Dynamic environment

A dynamic environment is one that has other processes operating on it, and which hence changes in ways beyond the agent's control.

Based on an output generated from static inputs in a specific moment we have classified this system as a static environment.
The element considered that would cause to have a dynamic environment are the following:

- Inputs values could be changed during the execution of the system
- The user agent have to observe the environment at the same time he is deciding the course of action

3.5. Discrete vs. Continuous environment

On one hand, a discrete environment has fixed locations or time intervals. On the other hand, a continuous environment could be measured quantitatively to any level of precision.
It is assumed that our system is following continuous behavior. This assumption is based on the fact that the input values are real numbers and they are not limited to any quantization type.

4. Agents architecture

In this section, the type of architecture designed to each agent will be defined.

The concept of architecture proposes a particular methodology for building an autonomous agent. The type of architecture defines the way in which the construction of the agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact.

The different types of agent architectures taken into account are the following:

- **Reactive:** An agent with a reactive architecture has to be able to react in an appropriate way to the dynamic changes in its "environment". In Figure 2, a basic schematic of this type of architecture is shown.

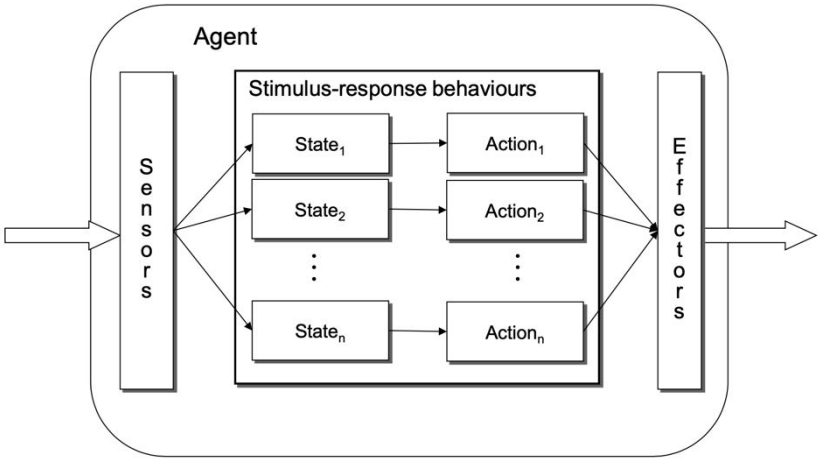

Figure 2. Basic schema of reactive architecture

- **Deliberative:** A deliberative agent has an explicit symbolic model of the world. Its decisions are made via logical reasoning, based on pattern matching and symbolic manipulation. In Figure 3, a basic schematic of this type of architecture is shown.

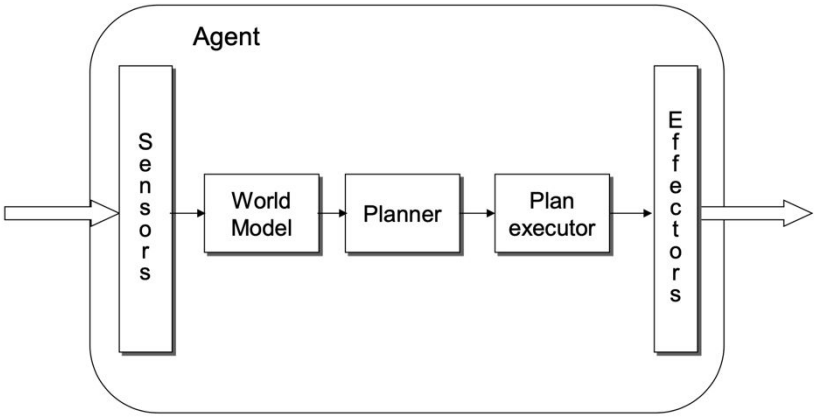

Figure 3. Basic schema of deliberative architecture

- **Hybrid:** A hybrid agent attempts to marry classical and alternative approaches. In Figure 4, a basic schematic of this type of architecture is shown.

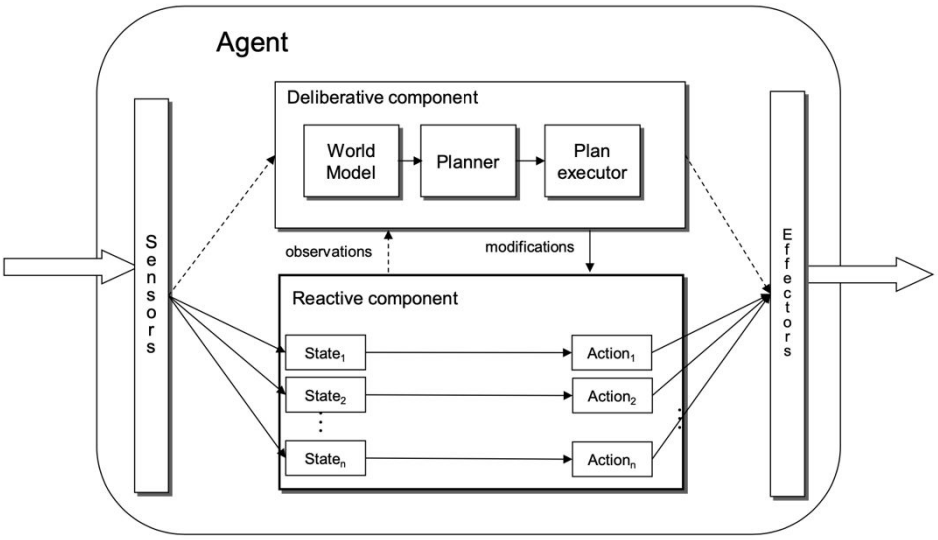

Figure 4. Basic schema of hybrid architecture

4.1. User agent

The goal of the user agent is to capture or receive data from the environment, from sensors or from user input and provide that information to the manager agent. Also, it will be responsible for getting the query response from the manager agent and display it to the user.
Because the user agent is sending and displaying information with very simple rules, the reactive architecture will suit it best.

4.2. Manager agent

The manager agent's task is to act as the main node of communications between the user agent and the fuzzy agents that will be created to perform tasks. The manager must perform both reactive and deliberative tasks. On the one hand, must perform communications between the agents, and there is no need for complex rules for that. On the other hand, If the manager agent has to store information on past data or has to solve conflicts between fuzzy agents, has to have deliberative functionalities added. For those reasons, a hybrid architecture will suit the agent best.

4.3. Fuzzy agent

Fuzzy agents are in charge of performing fuzzy logic in data provided by the manager agent as well as defuzzify the result.
We believe the most suitable architecture would be hybrid, as it needs a symbolic model of the world to infer the result asked by the manager agent thus needing deliberative functionalities as well as reactive functionalities to react when information is requested from the manager agent.

5. Agents properties

Each agent has a set of characteristics/properties provided by the author that are considered important to the notion of the agenthood.
The properties that an agent can contain are the following:

- **Flexibility:** To be flexible, an agent must be Reactive, Proactive and Social.
- **Reactivity:** There is a need to maintain an ongoing interaction with the agent's environment, and to respond to changes that occur in it (in time for the response to be useful).
- **Proactivity:** Generating and attempting to achieve goals/Behaviour not driven solely by events/Taking the initiative when appropriate/ Executing actions/giving advice/making recommendations/making suggestions without an explicit user request/Recognizing opportunities on the fly.
- **Social Ability:** It's the ability to interact with other agents (and possibly humans) via some kind of *agent-communication language*, and perhaps cooperate with others.
- **Rationality:** An agent will act in order to achieve its goals, it will not apply deductive procedures (blind triggering of rules) without a purpose.
- **Reasoning capabilities:** This is the characteristic that distinguishes an intelligent agent from a more "robotic" reactive-like agent (includes fuzzy logic).
- **Learning:** It can be considered as the [automatic]improvement of the performance of the agent over time.
- **Autonomy:** Ability to pursue goals in an autonomous way, without direct continuous interaction/commands from the user.
- **Temporal continuity:** Agents are continuously running processes.
- **Mobility:** Mobile agents can be executing in a given computer and, at some point in time, move physically through a network (e.g. Internet) to another computer, and continue their execution there.

5.1. User agent

Considering the reactive architecture of this agent, the properties of this type of agent are the following:

- **Temporal continuity:** It must be available at all times to get and deliver current information about the environment observations and status.
- **Reactivity:** It is the first and only responsible for interacting with the environment.
- **Social:** The user agent will interact with the user in order to receive the information and the manager agent to handle the request.
- **Rationality:** It will only extract the information about the environment if and only if it is asked to do, as it has no need to extract data if no other agent (mainly the manager agent) has requested it.

5.2. Manager agent

Considering the hybrid architecture of this agent, the properties of this type of agent are the following:

- **Reactive:** This agent must be able to act when receives a request.
- **Temporal continuity:** As long as the system is up, the manager agent has to be running as it is the responsible for the communications in the system.
- **Social:** The agent is responsible for the communication between all fuzzy agents and the user agent with itself.
- **Rationality:** The manager agent will ask for data or will create an agent only when a task has to be fulfilled. Otherwise, there is no point in asking for it.
- **Proactive:** It has to be able to know when is the best moment to perform certain actions as well as perform actions asynchronously to the user agent's data or the fuzzy agents messages.
- **Reasoning Capabilities:** This agent must be able to solve conflicts if needed, so it has to be able to reason.
- **Proactive:** It has to be able to know when is the best moment to perform certain actions as well as perform actions asynchronously to the user agent's data or the fuzzy agents messages.
- **Flexibility:** It is reactive, social and proactive, hence we can consider Manager Agent is flexible.

5.3. Fuzzy agent

Considering the hybrid architecture of this agent, the properties of this type of agent are the following:

- **Reactivity:** This agent acts in the environment depending on the request received.
- **Social:** This agent has the ability to communicate with other agents.
- **Rationality:** The fuzzy agent will perform the necessary actions in order to fulfil the requested action.
- **Reasoning Capabilities:** By definition, this agent must be able to perform fuzzy logic on the data provided by the manager agent.

All of them will be benevolent and veracious, since they will always try to do what they are asked for, and they won’t communicate false information.

6. References

[1] Glavic, Mevludin (2006). Agents and Multi-Agent Systems: A Short Introduction for Power Engineers. Retrieved 17:36, October 16, 2020, from https://people.montefiore.uliege.be/glavic/MAS-Intro_Tech_report.pdf

[2] Rocha, Jorge (2017, 28 June). Introductory Chapter: Multi-Agent Systems. Retrieved 21:05, October 16, 2020, from https://www.intechopen.com/books/multi-agent-systems/introductory-chapter-multi-agent-systems

[3] Boissier, Olivier. An Introduction to Multi-Agent Systems. Retrieved 20:05, October 16, 2020, from https://www.intechopen.com/books/multi-agent-systems/introductory-chapter-multi-agent-systems