Agent-Oriented Software Engineering
2019/2020
Project Report

Francesco Penasa 215038

August 20, 2020

1 Introduction

In this report we disclose the second assignment of the AOSE course which consists of designing the management of an autonomous warehouse, where drones and robots are employed to retrieve, move and deliver packages.

The designed Multi-Agent System is completely autonomous, there is no need for any run-time human interventation. The implementation of such system has been done following an agent-oriented approach. There are no "hard-coded" solutions and the delivered solution works equally well on multiple computers.

No `C#` scripts have been modified or added to the original implementation, we worked exclusively on the Prolog files and such files are send as attachment to this report.

2 Solution

The Prolog files are designed for one artifact (*Box*) and five agents (*GameManager, PickupArea, Drone, RailBot, SortingBot*), in this section we will inspect every Prolog file in the details in the chronological order used to implement them to observe the implementation choiches that have been made.

Box It has not been modified, no changes were needed.

GameManager During the development of the other agents `GameManager.prolog` have been modified to generate only two boxes, in this way, less actions were showed on screen and it was easier to debug the problems emerged from the code.

In the end the *GameManager* agent have been restored to its original implementation in order to check if the implementation of the other agents were correct when multiple interactions were made.

PickupArea We added to `PickupArea.prolog` some plans to call a drone whenever a box spawn and to destroy a box whenever such box was successfully delivered.

Drone `Drone.prolog` is one of the most used agents since it interacts both with *PickupArea* and *RailBot*, it is used to move boxes from the *PickupAreas* to the *RailBot* platforms and viceversa. Some plans have been added to execute the following actions.

- Pickup boxes from the *PickupArea* and bring them to the *RailBot*. (`pickupBox(B)`, `send_to_RailBot(B)`)
- Return to the charging station after each flight. (`goto_chargin_station()`)
- Wake-up the *RailBot*. (`call_railbot(B)`)
- Deliver boxes from the *RailBot* to the correct *PickupArea*. (`toDelivery(B)`, `deliver(B)`)

RailBot `RailBot.prolog` has the assignment to move the boxes between the Exchange Area and the RailBot platforms, it interacts with *Drones* and *SortingBot*. Some plans have been added to handle the following actions.

- Check if a box is supposed to be moved to the exchange area or not. (`check_box(B)`, `leaveBox(B)`)
- Move a box to the exchange area. (`pickupBox(B)`)
- Move a box from the exchange area to the correct platform. (`pickupBox(B)`)
- Wake up the *SortingBot* if a box is in the exchange area waiting to be sorted. (`call_SortingBot(B)`)
- Wake up a *Drone* if a box is in a platform waiting to be delivered. (`call_Drone(B)`)
- Return to the charging station after each movement. (`goto_charging_station()`)

SortingBot `SortingBot.prolog` is responsible for the sorting of the boxes from an exchange area to another one and it interacts with *RailBots*. The following actions are the ones allowed by the added plan.

- Pickup a box from an Exchange Area and move it to another one. (`sortBox(B)`)
- Wake-up a *RailBot*. (`call_railbot(B)`)
- Return to the charging station after each action. (`goto_charging_station(B)`)

3 Encountered problems

The main issue encountered during the project development has been the interaction between the agents, some of the first solutions developed would lead the agents in a deadlock state. To resolve this problem an extensive use of belief have been done, to achieve an asynchronous execution of all the agents.

Another issue has been the peculiar syntax of Unity Prolog and the fact that since it is a very recent technology there are no syntax highlights tools for Prolog to helpl with the debug of syntax errors.

4 Conclusion

The lab sessions have been essentials in order to gain the knowledge required to complete this assignment and I found them extremely interestings, not only because I'm particularly interested in this topic but also because such sessions are practice-oriented and give the student an insight on the applications of this subject.