AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS (AASMA)

ACADEMIC YEAR - 2017/2018

LAB CLASS 3 – DELIBERATIVE AGENTS IN THE LOADING DOCKS SCENARIO

1. GOALS

1. To develop the behaviors of the robots in the *Loading Docks* scenario, using a deliberative approach.

2. THE DELIBERATIVE AGENT

The following pseudo-code presents the deliberation procedure of a deliberative agent (BDI).

```
B <- B0 /*Initial beliefs*/
    I <- I0 /*initial intentions*/
    while true do
        P <- get-perceptions
        B <- beliefs-revision-function(B,P)
        D <- option(B,I)
        I <- filter(B,D,I)
        plan <- build-plan(B,I,Ac)
        while (not (empty(plan) or succeeded(I,B) or impossible(I,B))) do
            instruction <- head(plan)
            executa(instruction)
            plan <- tail(plan)
            P <- get-perceptions
            B <- beliefs-revision-function(B,P)
        end-while
    end-while
```

2.1 LOADING DOCKS: BELIEFS, DESIRES AND INTENTIONS

The beliefs of an agent in the *Loading Docks* environment include all the information captured by its perceptions.

Desires include all the possible options for the agent to take regarding its perceptions. However, in the *Loading Docks* environment, the three possible desires occur independently.

- Grab – The agent wants to grab a box
- Drop – The agent wants to drop the box
- Initial-position – The agent believes that all boxes had been delivered and want to return to its initial-position

The intention will reflect the chosen desire to perform. There are also three intentions in the *Loading Docks* scenario, one for each desire.

3. PROVIDED FILE (NETLOGO)

- **`robot-loop`** – This procedure contains the BDI approach presented in Section 2. However, the implementation of **`BDI-options`**, **`BDI-filter`** and **`build-plan-for-intention`** is missing, which causes the agents to initially behave using a reactive approach.
- Beliefs – The agents internal state contains a map of the warehouse where the information about the ramp and shelves is updated during their navigation in the warehouse.
- Desires – Since in this scenario the agents only have one desire at a time, it is represented as the string **“`grab`”**, **“`drop`”** or **“`initial-position`”.**
- Intentions – They are based on desires and determine the agent’s behavior. Intentions are represented as the tuple <desire, position, heading> in order to specify the position and heading that allow the agent to reach the desire. For instance, the chosen desire of grabbing a box will generate an intention with the position and heading required to grab a specific box.
- Plans – They contain the set of instructions to achieve the current intention.
  - Plan-intructions – There are four plan-instructions in the provided file:
    - **`build-instruction-find-adjacent-position`**
    - **`build-instruction-find-heading`**
    - **`build-instruction-drop`**
    - **`build-instruction-grab`**
- Communications – The following procedures allow communication between the agents:
  - **`send-message`**
  - **`send-message-to-robot`**
  - **`new-message`**
- Supplementary procedures – Besides all the provided procedures to create, access and modify the provided internal abstract types, there are other useful procedures:
  - **`adjacent-position-of-occupied-ramp`**
  - **`adjacent-position-of-free-shelf`**
  - **`build-path-plan`**

4. Exercises

4.1 Agent behavior

In the previous class, the reactive agent architecture revealed to be insufficient. Although the agents were able to delivered all boxes in the correct shelves, they were not able to return to their initial positions. In this class, the agents can keep record of warehouse map and, therefore, more efficient solutions might be created. Elaborate a strategy to create that solution and assume the agents know the total number of boxes.

Note: Do not call the function delivered-boxes! (Why?)

4.2 BDI Agent

Implement the deliberative agent to solve the *Loading Docks* problem, based on beliefs, desires and intentions (BDI). In order to do that, implement the procedures **`BDI-options`**, **`BDI-filter`** and **`build-plan-for-intention`**. The first one updates the agent's current desire, while the second convert it into an intention. The third one builds a plan to achieve the current intention.

4.3 Commitment

Implement persistent intentions, in a way that after an intention has failed, the agents try to find an alternative. For instance, during the executions of two plans, if two agents have collided, they should find another way to achieve the same intention. Use the procedure **`collided`** to solve that problem.

4.4 Problem

Discuss some problems of the developed deliberative agents.

[WOOLDRIDGE02] - WOOLDRIDGE, M.; AN INTRODUCTION TO MULTIAGENT SYSTEMS; JOHN WILEY & SONS, LTD; 2002