

ID2209 - Distributed Artificial Intelligence and Intelligent Agents

Assignment 2 - Negotiation and Communication

Kishore Kumar, kishorek@kth.se

Abel Valko, valko@kth.se

November 2021

1 Introduction

The objective of this assignment was to simulate a dutch auction between an auctioneer and multiple participants with communication implemented through FIPA protocol.

1.1 How To Run

Open Gama and import the models found in folder "Lab 1". The experiments can be run by clicking the FestivalExperiment button. The model can be tweaked using the following parameters:

- *numberOfAuctioneers* and *numberOfParticipants* control the number of created agents of each type
- *preferredPrices* are a list of preferred prices for participants, deterministic for reproducibility.
- The variables *maxAuctionCycles*, *initialOffer*, and *lowestPrice* determine the auctioneers price over the auction cycles.

2 Species

2.1 Auctioneer

The auctioneer is the agent in who leads the auction. This agent possesses the *fipa* skill, the parameters as mentioned above, and the following reflexes:

- *sendRequest* sends an inform message to all participants informing them that the auction is starting.
- *initiate_cfp* initiates the auction by sending a cfp with information about the auction and the current price to all agents.
- *refusals_handler* receives the refusals from the participants if no offers (proposes) have been made. It determines the next price offer and sends a new cfp to the participants if that price is above the minimum price threshold for the auctioneer, otherwise the auction ends.
- *proposals_handler* receives proposals from participants, accepts one proposal, and rejects the rest.

2.2 Participant

The participants are the agents that are taking part in the auction, aiming to buy the item on sale. For this first part it is assumed that all participants are interested in the item for sale. They have a preferred price as mentioned above which they are willing to pay for the item. They also have the following reflexes.

- *receive_inform_messages* receives the inform message that the auction is starting.
- *respond_to_proposals* receives cfps from the auctioneer and replies either with a *propose* or a *refuse*, depending on if the price suggested is lower or higher than the participant's preferred price respectively.
- *receive_reject_proposals* receives *reject_proposals* from the auctioneer.
- *receive_accept_proposals* receives *accept_proposals* from the auctioneer.

3 Results

The printout for the implemented auction simulation can be seen below. The first few timesteps can be seen in Figure 1 for five participants. At timestep 1 the auctioneer sends an inform message to all participants who receive it at time 2 as the auctioneer sends the first cfp with the starting price.

```
Simulation started with 5 participants and 1 auctioneers
Auctioneer0: Starting auction
(Time 2.0): Participant0 receives inform messages
    Participant0 receives a inform message from Auctioneer0 with content ['start_auction','Merch']
(Time 2.0): Participant1 receives inform messages
    Participant1 receives a inform message from Auctioneer0 with content ['start_auction','Merch']
(Time 2.0): Participant2 receives inform messages
    Participant2 receives a inform message from Auctioneer0 with content ['start_auction','Merch']
(Time 2.0): Participant3 receives inform messages
    Participant3 receives a inform message from Auctioneer0 with content ['start_auction','Merch']
(Time 2.0): Participant4 receives inform messages
    Participant4 receives a inform message from Auctioneer0 with content ['start_auction','Merch']
(Time 2.0): Auctioneer0 sends a cfp message to all participants
Initiating for the price:500.0
(Time 3.0): Participant0 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',500.0]
Price is too high, rejecting
(Time 3.0): Participant1 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',500.0]
Price is too high, rejecting
(Time 3.0): Participant2 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',500.0]
Price is too high, rejecting
(Time 3.0): Participant3 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',500.0]
Price is too high, rejecting
(Time 3.0): Participant4 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',500.0]
Price is too high, rejecting
(Time 3.0): Auctioneer0 no proposals
    Auctioneer0 receives a refusal message from Participant0 with content ['Reject']
    Auctioneer0 receives a refusal message from Participant1 with content ['Reject']
    Auctioneer0 receives a refusal message from Participant2 with content ['Reject']
    Auctioneer0 receives a refusal message from Participant3 with content ['Reject']
    Auctioneer0 receives a refusal message from Participant4 with content ['Reject']
New proposal for the price:440.0
```

Figure 1: Printout of the start of the auction.

All participants reject this price and so the auctioneer sends another cfp with a lower price. This continues until the price is sufficiently low. At this point, as can be seen in Figure 2, two participants reply with a proposal to the cfp. The auctioneer then accepts one of them and rejects the other.

```
(Time 5.0): Participant0 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',380.0]
I am buying it for 380
(Time 5.0): Participant1 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',380.0]
I am buying it for 380
(Time 5.0): Participant2 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',380.0]
Price is too high, rejecting
(Time 5.0): Participant3 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',380.0]
Price is too high, rejecting
(Time 5.0): Participant4 receives a cfp message from Auctioneer0 with content ['Propose_price','Merch',380.0]
Price is too high, rejecting
(Time 5.0): Auctioneer0 received proposals
    Auctioneer0 receives a proposal message from Participant0 with content ['Accept']
    Auctioneer0 striking a deal with Participant0
    Auctioneer0 receives a proposal message from Participant1 with content ['Accept']
    Auctioneer0 rejecting the deal withParticipant1
(Time 6.0): Participant0 received an accept_proposal message from Auctioneer0 with content ['Deal!!!']
(Time 6.0): Participant1 received a reject_proposal message from Auctioneer0 with content ['Already struck a deal!!!']
```

Figure 2: Printout of the end of the auction.

4 Discussion

The assignment was implemented successfully and gave insights on both the dutch auction itself as well as the fipa protocol. However, without sufficient documentation from GAMA it was rather difficult to determine what protocols were valid and how conversations should be handled. This caused some frustration and required a lot of time in debugging without contributing much to an understanding of the focus of the assignment. Despite this, the lab gave a very good picture of how more complicated systems could be built using similar communication protocols.