

# ID 2209 Distributed AI and Intelligent Agents : Homework 2

avneesh@kth.se, gaiddon@kth.se (Group 6)

November 22, 2016

## Task 1

### Introduction

The objective of the assignment was to implement FIPA Dutch Auction interaction Protocol for smart museum consisting of two different entities:

- Artist Manager Agent representing the auctioneer
- Curator Agent representing a bidder

In this scenario we instantiate 4 agents, 1 Artist Manager and 3 Curators which are the bidders for the auction.

JADE, an open source platform for peer-to-peer agent platform was the given framework for developing the aforementioned agents.

### Design Notes

1. The 3 entities; Profiler, Tour Guide and Curator are implemented as JADE agents by inheriting from `jade.core.Agent` class.
2. All agents register to JADE's yellow pages service (DFService Class) as soon as they start-up.
3. Curator Agents registers Bidding service in DF and Artist Manager Agent subscribes itself to this service to find potential bidders.
4. Artist Manager INFORMS all the bidders the auction is starting and sends periodically a CFP message with the current price which is decreasing each new iteration until a reserve value.
5. Curator Agent makes a proposal if the current price corresponds to its own price. Otherwise it waits for a new price from the Artist Manager.
6. Artist Manager accepts proposal from Curator if the proposed price corresponds to its current price. Otherwise it rejects this proposal.

7. Once the Artist Manager has reached its reserve price, the auction ends without winner.
8. When the auction is finished, Artist Manager INFORMS all the bidders.

### Instruction to run the program

1. Download the source code and create an eclipse project (or net-beans).
2. The system depends on jade.jar and therefore add it to the project's build path as an external jar file.
3. Create a run configuration for the project with jade.Boot as main class and -gui as program arguments.
4. Run the program. This will launch JADE's Remote Agent Management GUI.
5. Select 'Main Container' in the left AgentPlatforms tree.
6. Go to Action Menu and choose 'Start New Agent' to start the four Agents with different local names.
  - a) Artist Manager : Choose class agent.ArtistManager. No argument is needed.
  - b) Curator Agent : Choose class agent.CuratorAgent and provide a preferred price for this agent.
7. In the console output, you can see the auction progress.

*Note: If you want more than one bidder for the auction you must start all the curators before the Artist Manager.*

### Conclusion

In the agent implementation, the following JADE provided behaviours were used :

- Simple Behaviour
- Sequential Behaviour, for scheduling behaviours of Artist Manager
- Parallel Behaviour (in the Artist Manager)
- OneShotBehaviour, for INFORM messages
- TickerBehaviour for performing CFP messages periodically
- CyclicBehaviour for listening constantly and handling replies

We could improve this FIPA Dutch Auction implementation by adding a Curator behaviour which would have different strategies to negotiate the price with the auctioneer.