

# Java-based simulation of agent behavior and transactions

CSC-40045: DISTRIBUTED INTELLIGENT SYSTEMS

ALEX FARRELL (15005594)

## Contents

Introduction .....	2
Buyer Agent.....	2
Seller Agent .....	2
Broker Agent (Transactions) .....	2
Conclusion.....	3

## Introduction

This coursework looks at the implementation of a multi-agent system using Java. The system is required to have 3 types of agent including a buyer, seller and broker agent. The buyer and seller agent handle all operations that these users would normally do. The broker agent handles all the transactions between the buyer and the seller. For proof of concept, the system is only programmed to deal with integer values.

## Buyer Agent

The buyer (or customer) agent handles the creation of the customer profile. This information includes the constraints that the customer has when searching for a product including the maximum price, quantity required and the type of delivery. Each of these constraints is programmed in its own method and is called by the constructor of the class. The data returned from each method is fed into a 1-dimensional integer array that stores the data for easy access by the broker agent.

The final method in this class is the method that selects the offer that the customer will choose. This method is implemented by feeding in the number of offers received from retailers and choosing one at random. A possible improvement to this method would be for the customer to look at the data in the offers received and rank them according to preference and make a more thought out choice.

## Seller Agent

The seller (or retailer) agent handles the creation of an individual retailer's stock information. The first method 'productId', used a random number generator to select 10 unique random numbers between 1 and 20. The numbers represent the products that the retailer stocks. Inside this method, there is a loop that loops over the array to see if the number chosen already exists and if this is the case, a new number is generated until a unique number is found. Like the buyer agent, each piece of information about a product is created in its own method. However, rather than returning the value the information is input directly into the 2-dimensional array, which is created at the top of the class as an instance variable.

## Broker Agent (Transactions)

The broker agent is where all the transactions are dealt with. At the start of the program, an array list is created which stores the stock information for all the retailers. This allows the retailer information to be easily extracted further on in the process. For proof of concept, a single customer has been created. The first stage utilises a method that loops through the stock information for each retailer to check if the product requested exists for that retailer and if this is the case, the retailer information is added to a new array list. Each time each of the constraints are analysed, a new array list is created until the final constraint when a final list of offers is created. If no offers are received, the customer is informed of this. A possible improvement to this would be for the customer to review their request and make alterations. This could be further improved if the retailers who stock the requested product, inform the customer of their stock status and other information so the customer can make a decision.

Finally, the offers received (if any) are shown to the customer, who then chooses which one they want to go for.

## Conclusion

To conclude, there are multiple improvements that could be made to the implementation of this multi-agent system, however as showing proof of concept, the process in which is followed to reach the final outcome, shows the flow of information and communication between different agents to reach a common goal.