Multi Agent Systems Coursework Report

# Roderick Ewles 40330977

## Introduction

## Model Design

The model is designed with 5 agents, those being: Day Ticker, Customer, Manufacturer, Expensive Supplier and Cheap Supplier. An outline of these agents, their responsibilities, communication and the ontology will be given below.

### Day Ticker Agent

The day ticker agent tracks the days as they pass in the simulation and makes sure that the other agents are synchronised with this. On setup the agent registers with the “yellow pages as the day ticker”. The agent has a case switch which controls its function on each day. The first case statement is where the agent finds the other agents using the yellow pages and stores them in an array list used throughout the subsequent statements. It then informs all of the simulation agents that a new day is beginning, following which the step for the case statement is incremented.

The second stage of the case statement waits for the other simulation agents to inform it that they have finished their activities for the day after which the step is incremented again. When the step increments to 2 the case step variable is reset to 0 so long as there are days left in the simulation. If the hundred days are simulated the day ticker sends a terminate instruction to the other agents in the system. Both of the messages sent by the day ticker are inform messages as they tell the other agents either that it is a new day or that the simulation is over, and they should terminate.

### Customer Agent

The Customer agent sends the manufacturer agent one order every day. It does this in the method specified in the coursework brief. On setup the customer agent registers with the yellow pages. Everyday the Customer agent follows a cyclic behaviour called “Ticker Waiter” which has several sub-behaviours. This will either follow the sequence of sub behaviours if a “new day” message is received from the day ticker or terminate the agent if a terminate message is received. The main responsibility of the customer agent is to generate a new supplier order on a new day.

### Manufacturer Agent

The Manufacturer agent carries out the majority of the work. In the Setup method the build schedule is initialised, the daily production capacity is set and the agent registers with the yellow pages. In addition, a ticker waiter behaviour added to the agent similar to the Customer agent. The first two sub behaviours allow the manufacturer to find Customers and suppliers. Following this the sub-behaviours are added that allow the Manufacturer to decide whether to fulfil orders, schedule them, send supplier orders and build them when scheduled. This will be discussed in more detail later in the report. Essentially the Manufacturer’s main responsibility is to decide whether to accept orders then fulfil them at the correct time. The Manufacturer also has an instance of a warehouse class that tracks the stock and has a method to calculate the storage cost if parts are kept overnight, this is used in the profit calculation to evaluate how efficient the manufacturer is.

### Supplier Agents

The supplier agents function similarly, and essentially send the manufacturer parts with differing delays. They wait for orders from the supplier on each day until they receive an ordering done Supplier Order.

### Ontology Design

The ontology uses one predicate owns. The actions in the ontology are the orders which are used and tracked by manufacturers and suppliers. The main difference between the two order types being that Customer Orders are requests that are tracked and supplier orders have the additional use of requesting that the supplier stop supplying the manufacturer for the day, the supplier agents will wait until they receive this special order type. The Concepts are initially defined as items which have a price, all other phone parts inherit from this.

## Model Implementation

### Customer agent

The behaviours of the customer agent are added in its ticker waiter behaviour. The first sequential behaviour of “ticker waiter” is “find manufacturer” where the agent uses the yellow pages to find the Manufacturer. The second sub-behaviour is “generate order”. In this sub-behaviour the new Customer Order is generated. The Customer Order is very important as it is the part of the ontology that the manufacturer uses to decide whether to track as well as being tracked within the manufacturer to establish when accepted orders have been completed. Aside from specifying the components, quantity unit price, day due and the late penalty, it also has values for accepted, gross profit, net profit, net cost, net profit, fast turnaround, order id and scheduled. The second set of values is used by the manufacturer aside from order id which is set by the customer and used for tracking. The final sub-behaviour is an end day behaviour which informs the day ticker agent that the day is over.

### Manufacturer Agent

The behaviours of the manufacturer agent are added in the ticker waiter behaviour similarly to the customer agent. The first two of these allow the manufacturer to find suppliers and customers. Following this there is an order handler behaviour that listens for customer orders. This behaviour will wait until an order has been received from each customer. In addition to this this behaviour also handles supplier orders being received and updates the warehouse stock as appropriate. Following this there is an order decider behaviour that decides whether to accept orders based on if they can be fulfilled in time as a just in time (JIT) approach to order handling has been adopted. These orders are then checked against the manufacturer’s production capability to see if they can be scheduled appropriately. Following this an order sender behaviour is used to schedule the orders so that they arrive on the day required as dictated by the production schedule. There is then a pair of supplier order generator behaviours used to send the orders to the relevant supplier based on the supplier schedule. After this the next behaviour is the build behaviour which checks what orders are scheduled to be built that day and then subsequently checks if the necessary parts are in stock, if so, the order is fulfilled. Finally, there is an end day behaviour similar to the customer agent.

### Supplier Agents

Again, both supplier agents have a ticker waiter behaviour with several sub-behaviours. The first of these is a find manufacturer behaviour that use the yellow pages to find the manufacturer. The next behaviour is a Fulfil order behaviour that sends the fulfilled supplier order to the manufacturer based on the order schedule. This is where the two supplier types differentiate as the expensive supplier fulfils received orders on the next day, the cheap supplier doing the same after four days. Following this the next sequential behaviour is the order handler that receives supplier orders until a finisher order is received at which point the behaviour ends. The final behaviour is an end day behaviour that functions in the same way as the other end day behaviours.

### Constraints Implementation

The first constraint is enforced in the customer and manufacturer as well as the ontology due to the mandatory slots imposed on the customer order. Each phone must have certain parts, some being set in the manufacturer based on whether the supplier order is sent to either the cheap of expensive supplier at which point the remaining part prices are set. The delivery time constraints are enforced in the respective supplier types order schedules. The warehouse class fulfils the third constraint. The fourth constraint is enforced in the manufacturer’s build behaviour, as is the daily production capacity. The penalties for late delivery are also enforced in the manufacturer but are never really utilised due to how the manufacturer functions, similar to the warehouse cost. The final constraint is met in the manufacturers end day behaviour.

## Design of Manufacturer Control Agent Strategy

## Experimental Results

## Conclusion

## References

## Appendix 1: Ontology

## Appendix 2: Communication Protocols

## Appendix 3: Source Code