Agent-based Modeling and Simulation for an Order-To-Cash Process using NetLogo

John Villarraga[1], Kathleen M. Carley[2], John Wassick[3], and Nikolaos Sahinidis[1]

August 1, 2017
CMU-ISR-17-113

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3]The Dow Chemical Company, Midland, MI 48674, USA

Additional support was provided by the center for Computational Analysis of Social and Organizational Systems and the Institute for Software Research at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies. We are grateful to John Wassick from DOW Chemical Company for his support and advice.

Center for the Computational Analysis of Social and Organizational Systems
CASOS technical report.

Keywords: supply-chain, agent based modeling, NetLogo,

Abstract

This paper presents a study on the Order-To-Cash process of a supply chain using agent-based modeling. Supply chains are composed of multiple decision makers where each of them play an important role in the entire system. If one of the variables controlled by these entities malfunctions, this can significantly increase a customer's order fulfillment time. This study investigates organizational issues of a supply-chain and presents solutions by conducting a variety of experiments using NetLogo. Results show that the performance of the model is affected by a large volume of orders due to errors and exceptions that occur throughout the simulation. We also found that our system contains bottlenecks that cause significant amount of delays in the model. Our analysis demonstrates that in some scenarios hiring new individuals where there is a bottleneck could greatly increase the efficiency of the model. Moreover, hiring new people on the same role proved to be more relevant than investing in training of individuals as the agents of the model improve their efficiency in the simulation run.

Table of Contents

1. Introduction

The Order-To-Cash (OTC) process is composed of a network of decision makers containing customer representatives, retailers, planners, transportation specialists, carriers, manufacturing plants, distribution centers and others. These individuals/entities share information with one another to manufacture and deliver goods to customers based on time sensitive orders. The workflow of the process from where the customer creates an order to the fi point where the product is made and delivered to the customer is driven by a large number of variables. To coordinate and predict how long it will take for a customer to receive an order is always difficult as distribution centers can run out of material during the process, manufacturing plants can have a failure in their system, workers can become ill, transportation delays can occur and so forth.

Modeling the Order-To-Cash process is extremely complex due the number of entities and variables that are involved in completing an order successfully. The biggest problem facing the supply chain industry is the lack of research on dynamic networks and automation of their data interchange [1, 2, 3]. The nature of this process hinders the use of mathematical equations to model each entity or even steps of the entire process as the complexity of the system imposes a huge computational cost for solving the problem [4, 5]. However, since the rise of object-oriented programming (OOP), new modeling techniques have evolved that allow us to model complex dynamic systems [6]. One of the most popular techniques is agent-based modeling (ABM), where multiple entities of a system are modeled individually but simulated together to study the emergent pattern that comes from the simulation by changing multiple input parameters. The flexibility of this modeling allows researchers to get the major inputs and outputs that will be affected by policy decisions such that you can make what-if types of analysis on the model.

Simulation has long been used to assess organizational designs (e.g., [7, 8]). Most of the time such models, are a theoretical and aimed at organizational processes in the abstract (e.g., [9, 10, 11]). Less common, though not rate, is the modeling of specific actual organizations. In this latter case, simulation is often used to look for backlogs [6] or to understand the impact of the informal and forma social network on organizational outcomes (e.g., [12]). Simulation has been found to be one of the most suitable techniques for capturing the dynamics of a supply chain [13]. However, a key issue in simulation modeling of complex systems is validation. Numerous validation strategies have been developed, such as comparison of simulated and real data results (e.g., [14, 15]), grounding using cognitive work analysis [16], docking [17], or face validation [18]. Herein, we employ face validation and validation in parts (where the input and the model are validated against the actual organization).

A variety of software packages have been developed to facilitate the implementation of agent- based modeling for complex dynamical systems. One toolkit that receives a lot attention within this community is NetLogo [19]. This is an agent-based programming language that allows researchers to build models and explore their emergent patterns through simulations [20, 21]. NetLogo provides a graphical user interface that facilitates the creation of agents and their environment [22]. It also provides a parameter sweep feature called BehaviorSpace used to conduct virtual experiments and generate data by simulating the model.

The goal of this project is to use intelligent agents to model the OTC process using NetLogo. After that, we will conduct virtual experiments to analyze emergent behaviors that arise by simulating the agent-based model. Intelligent agents will be responsible for one or more tasks in the process and each will interact with other agents to complete their assigned work. Most of the agents in this model transfer information to other agents in their network in order to complete the order requests made by a customer. Each agent is responsible for a diff t portion of the process, which allows them to operate in parallel or sequentially, if substeps are required for them to execute their responsibilities. One of the tasks of these agents is to process each order request in their queues in a FIFO (first-in-first-out) manner. We have also incorporated into the model levels of expertise for the agents. This attribute allows the agents to become more efficient as they process more orders. Even though people's behaviors in an organization vary from role to role, we have tried to model the agents representing people in each role generically, to reduce variability in the process.

This paper investigates the construction of an agent-based model for the OTC process at the Dow Chemical Company, but it is simplified from the overall system and so represents an abstraction of the organization. Section 2 reviews a number of literature papers that focus on agent-oriented supply chain frameworks. Section 3 presents the methods and assumptions used to build our agent-based model. Section 4 describes the virtual experiments conducted, and their results are explained in Section 5. The report concludes with Section 6 which describes the conclusions and future work.

2. Related Works

In this study, the process to be simulated using agent-based modeling is a network of actors of a supply chain from the Dow Chemical Company. The workflow of the interactions between the agents was provided by the company, but due to time constraints, only a simplified version of the supply-chain has been modeled in this work. There has been a variety of studies that use different approaches for analyzing networks of people in a supply-chain through agent-based modeling.

In a research study conducted by Akkermans, he built a model for a supply chain network with 100 agents composed of suppliers and customers [23]. Each agent carried a mental model of other agents in their network to adapt their behavior based on the performance of the agents they were interacting with [23]. Akkermans' goal was to fi any emergent supply chain net- works based on short-term and long-term performance between the agents in the network during a time span of 9 years or 450 weeks. The results of the simulation showed that the networks which prioritized short-term performance did better than the long-term oriented networks, as customers preferred the most recent performance of the supplier agents and were biased towards them over time.

Other researchers have used agent-based modeling to study the impact of supply chain dynamics on plant operation and scheduling. García-Flores and Wang showed that supply chain behavior is sensible to plant production patterns [24]. Since manufacturing plants are not flexible and agile enough to respond to uncertainty of chain dynamics, they created a negotiation procedure between the agents to avoid delays in distribution and production. Chu and et.al presented a two-level agent based model for scheduling a network of batch processes [5]. Their results demonstrated that agent-based methods can be applied to complex large-scale problems containing various uncertainties [5].

In another study, a multi-agent system framework incorporating machine learning techniques was developed for a dynamic supply-chain network. The main goal of this research was to create a model that could dynamically adapt to its environment at any given point in time. The learning algorithm used was the C5.0 classifier as it has a low tendency to overfit and provides a good performance. The framework was tested with a two-stage supply chain where two different parts go into the assembly of each of the two products and a combination of suppliers could supply these parts [25]. Results showed that the model was able to route the orders automatically to the most appropriate supplier in real-time, which increased the number of delivered orders.

In a study similar to the previous work, Fox and et.al investigated the issues and solutions for constructing an agent-oriented software architecture. Their approach relies on the use of an agent building shell, providing generic, reusable, and guaranteed components and services for communicative-act-based communication, conversational coordination, role-based organization modeling and others [26]. The biggest issue they faced in constructing the software was coordinating the behavior of the agents among their counterparts. The stochastic events that occur in a supply chain can affect the actions of any of the actors involved in the network, which makes it difficult to coordinate. To overcome that problem they added knowledge, control and decision elements to their software.

The literature studies reviewed in this section contributed towards the development and understanding of the agent-based model constructed for this project. The agents that represent the dynamics of our supply chain network were defined based on information provided by the Dow Chemical Company. Some of their actions have been modified with the purpose of increasing the complexity of the model so as to obtain results similar to the ones provided by the company. To create the agent-based model many assumptions have been made. These assumptions will be discussed in the following sections.

3. Methods

3.1 Model Description

The agent-based model built in this research study resembles 3 types of entities of a supply-chain: people, units and information agents. People agents become active once a request is created and have similar behavior to people in those roles. Unit agents are actors that describe unit operations. Information agents contain the information that is passed around by the interaction between people and unit agents. These agents were selected based on Figure 1, which shows the main abstractions of the complex transactional processes required to fulfill customer needs at the Dow Chemical Company.

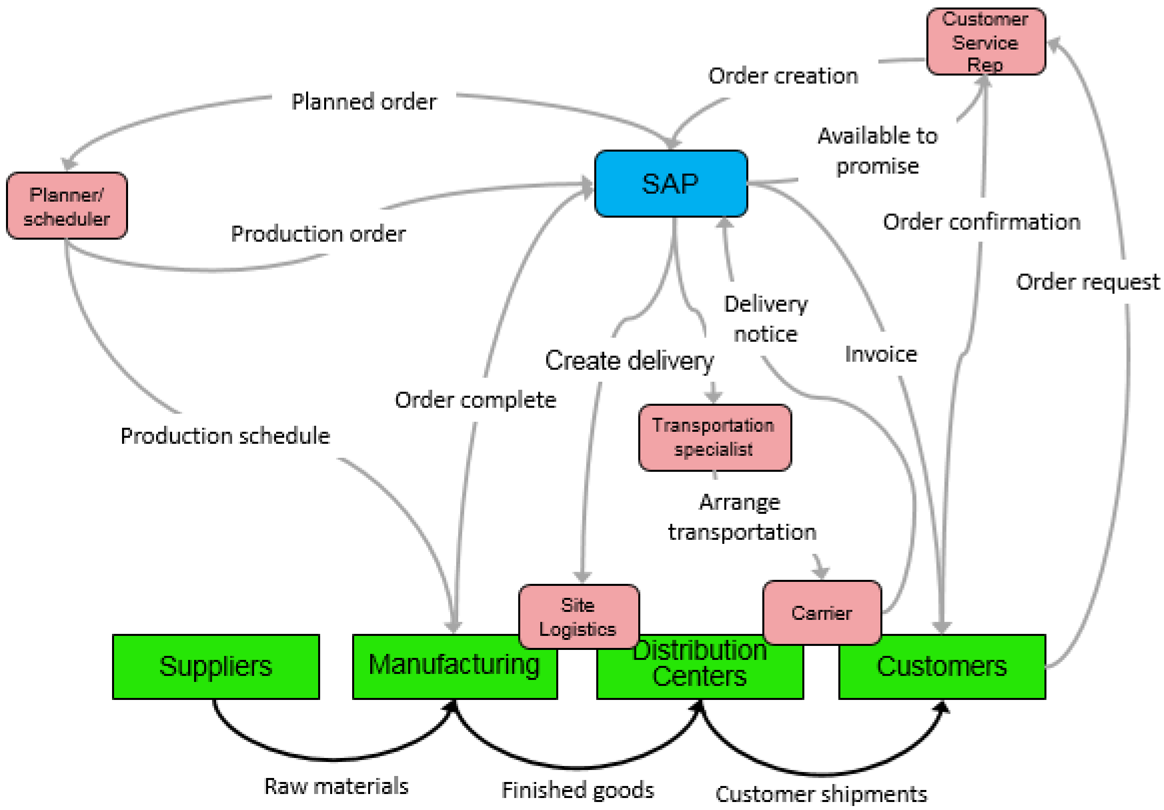


Figure 1: Simplified Order-To-Cash process provided by Dow Chemical

Our model contains 5 intelligent agents, which are the customer representative, planner/scheduler, distribution center, transportation specialist and carrier. Intelligent agents are those objects in the model that process data, store variables and have some impact on the overall process. The other agents are not classified as intelligent since they only move information around the process, and in some cases, they are the information itself. Therefore, they can be easily replaced by other agents. Below is a list of the agents in the model with their descriptions:

3.1.1 People agents

The agents described in this section represent people in a supply chain. Each of them have a queue, number of completed orders, level of expertise, and processing type. The only exception to the previous statement is the customer agent, which acts as an exogenous object.

1. Customer: Sends order requests to customer representative. Orders are sent every 10 ticks (ticks: built-in NetLogo variable that represents time).
2. Customer Representative (Rep): Receives request from customer and creates order in SAP.
3. Planner/Scheduler: Receives orders from SAP. Once it completes an order, assigns task to manufacturing plant and updates SAP.

4. Transportation Specialist (Spec): Receives orders from SAP. Once it completes an order, assigns task to carrier.
5. Carrier: Receives orders from transportation specialist. Its performance depends on the amount of inventory in distribution center. Once finished with its work, updates SAP.

3.1.2 Unit agents

1. Manufacturing Plant: Sends inventory to distribution center in a cycle manner. (This will be explained in section 3.2.2).
2. Distribution Center: Contains a certain amount of inventory that is used for every order in the process.

3.1.3 Information agents

Each order request in the simulation has different parts that need to be assigned to a respective agent. Make-to-order requests contain all the steps listed below, as these orders have to be processed from scratch. On the other hand, make-to-stock requests do not need to go through production and for this reason, all the information agents related to manufacturing are neglected for this type of request.

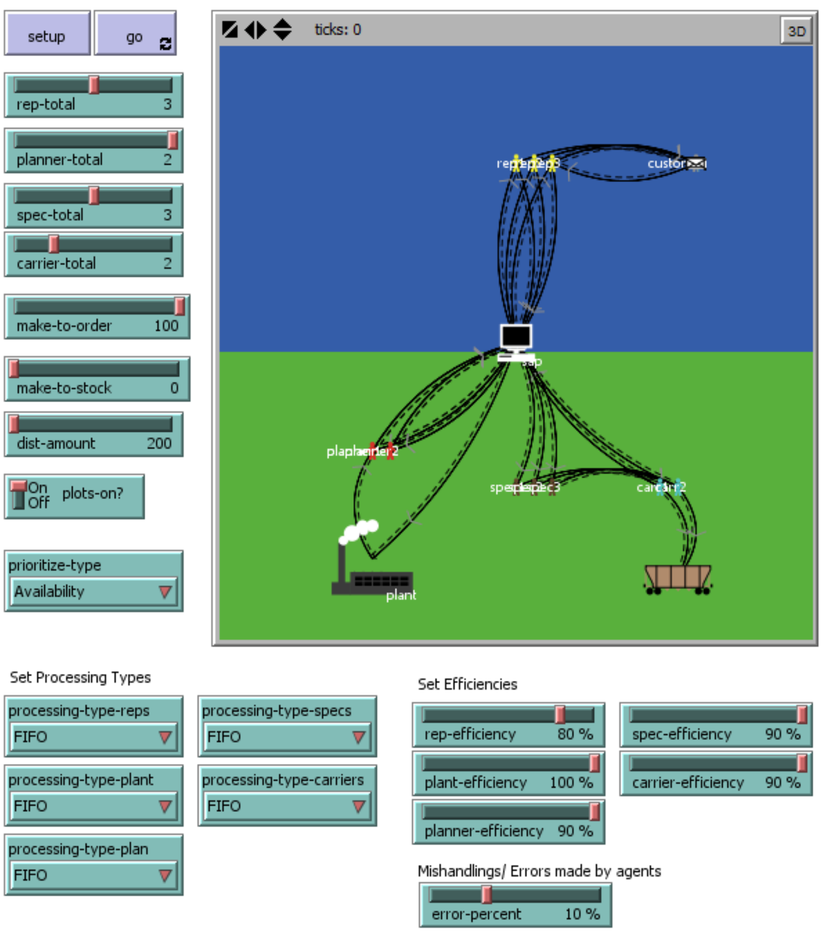
1. Order creation
2. Order request
3. Planned order
4. Production order
5. Production schedule
6. Create delivery
7. Arranged transportation
8. Delivery notice
9. Availability
10. Invoice
11. Order complete
12. Order confirmation

All the information agents contain the following variables:

- Work volume: Time required to process an order.
- Urgency: Rush order if above 90% (see 3.2.1).
- Incompleteness: Use to randomly select an exception (For more details see 3.2.7).

The SAP agent was modeled to reflect an enterprise resource planning (ERP) software that is used internally at the Dow Chemical Company. This system stores all the information about the orders and provides any updates about them to any actors involved in the process of completing an order. This agent is represented by using a computer graphic as shown in Figure 2. Once information is inserted into SAP, all the agents below it will obtain the information needed to initiate the completion of the request (refer to Figure 2).

Figure 2: Model with variables for virtual experiments



The model is built in NetLogo 5.3.1 as this is an agent-based programming language written in Java that contains many abstractions related to complex systems. The main reasons for using this language is its flexibility for relating agents to graphical representations, and its built-in parallelism.

3.2 Attributes of the Simulation

To initiate the simulation the user is allowed to select the number of orders to be processed for each type, the number of agents on each role, the processing type of the system, the initial efficiency of the agents, the prioritization criteria, and the chance of error (See Figure 2).

3.2.1 Type of order

The simulation supports two types of orders. These orders can be make-to-order or make-to-stock. Make-to-order will send the information agents through all the people and unit agents. While make-to-stock does not need to be processed by the planner and manufacturing plant.

All the orders in the model contain an urgency variable, which is utilized to distinguish between regular and rush orders. If the value of this variable is above 90, the request is a rush order and it will be given priority in the queue of the agents, neglecting any processing type.

3.2.2 Inventory cycle

We have created a cycle for the inventory in the distribution center. The manufacturing plant sends material to the warehouse in the amount of 1:5 ratio for small and large batches, respectively. This is mainly to avoid any exhaust of material in the distribution center as typically happens in a real manufacturing process.

3.2.3 Number of agents per role

The model also allows the user of the simulation to select the number of agents for each role. This attribute enables multiple case studies such as finding the point of delays in a network of people, and contrasting the performance of small and large organizations.

3.2.4 Queue management type

We have included a variety of processing types for the entire simulation or for individual agents. Based on the selection, the agents will use a different queue management strategy, which was motivated on the work of Law and et.al [18]. The simulation has the following type of strategies:

- FIFO: First-in first-out
- LIFO: Last-in first-out
- Rapid response: Process orders from smallest to largest
- Complicated: Process orders from largest to smallest

3.2.5 Initial efficiency

Each agent can be provided with an initial efficiency that will change throughout the simulation run. This feature determines the queue capacity of the agents. The table below provides details regarding efficiency levels:

Table 1: Expertise Table

Level of Expertise	Efficiency (%)	Queue Capacity	Completed Orders for Reward	Efficiency Reward
Low	0 - 64	10	10	20
Medium	65 - 84	20	30	10
High	85 - 100	40	60	5

To further clarify Table 1, if an agent with an initial efficiency of 70% completes 30 orders, its efficiency will increase to 80%. Then, the same agent will have to process 30 more orders to get an extra 5% of efficiency reward. For each simulation run, an agent could increase its efficiency by 35% if and only if its initial level of expertise was in the low efficiency range.

3.2.6 Prioritization criteria

Our model supports a prioritization criteria that can be either availability or expertise.

- Availability: The agents in the simulation will assign orders to whichever agent has vacancy in its queue.
- Expertise: The agents delegate the orders to the agent with the highest efficiency.

3.2.7 Errors/Exceptions

For the attributes to be explained in this section, we have applied some of the ideas of the virtual design team regarding exceptions and decision-making [6, 27]. The orders in the model have a chance for error that is defined at the beginning of the simulation. This error is used as a threshold to add a level of incompleteness to each request made by the customer. If an order is under that threshold, we modeled 3 different types of exceptions that occur after the order has been processed:

- Rework: The order is put back into the agent's queue with its original work volume [6].
- Partially correct: The order is put back into the agent's queue with half of its original work volume [6].
- Return: The order is returned to one of the agents that has completed a portion of the initial request.

These exceptions are determined randomly in the simulation. Also, depending on the level of incompleteness of the order, it can mutate into different exceptions.

4. Virtual Experiments

4.1 Small Organization

4.1.1 High-Demand Scenario

The experiments of this section were designed to find the agents that cause the greatest delays in a high-demand scenario. We have modeled a small supply chain that contains one agent in each role of the process. All the agents in the model will use FIFO as a processing type and have 90% efficiency, except for the manufacturing plant which we assume to be 100% efficient across all orders. In addition, we have included error rates in the simulation of 10% to consider mishandles and/or incomplete orders.

We decided to only create orders of the type: make-to-order, since these orders go through production and distribution using all agents in the model. The number of make-to-order requests will be increased from 10 - 300 to identify the points of delays.

Table 2: High Demand

Independent Variables	**Number of Test Cases**	**Values Used**
Make to order	30	10 - 300 in intervals of 10
Make to stock	-	0
Initial Inventory	-	200
# of Customer Representatives	-	1
# of Planners	-	1
# of Transportation Specialists	-	1
# of Carriers	-	1
Processing type	1	FIFO
Dependent Variables	**Number of Test Cases**	**Agent**
Average waiting time	-	customer reps, planners, specs, carriers
Average time of completed orders	-	-
Overall time	-	-

* The simulation will run 10 times for each test
* This is a 30-1 experimental design case

4.2 Large Organization Experiments

4.2.1 Expertise vs Availability

Once the bottleneck agents have been identified, we will increase the number of agents in those roles to analyze the optimal agents required to avoid delays. Now that we have multiple agents in the same role, we will explore what could be more effective to complete the orders, expertise or availability. Will giving the orders to the agent with the highest expertise decrease the completion time of the orders, or will assigning the order to the agent with the least number of orders on its queue be more efficient?

We could also conduct experiments to find when availability or expertise becomes important with increasing the number of orders. Is there a certain demand of orders where one should prioritize between expertise or availability?

Table 3: Expertise vs Availability

Independent Variables	Number of Test Cases	Values Used
Make to order	3	100, 200, 300
Make to stock	-	0
Initial Inventory	-	200
Bottleneck Agent	5	1 - 5
Initial Efficiency	-	90%
Prioritize-type	2	Expertise, Availability
Error rate	-	10%
Processing type	-	FIFO
Dependent Variables	**Number of Test Cases**	**Agent**
Average waiting time	-	Reps, planners, specs, carriers
Average time of completed orders	-	-
Total errors	-	Reworked, corrected, returned
Overall time	-	-

* The simulation will run 10 times for each test case
* This is a 5-3-2 experimental design case

4.2.2 Training vs New Hiring

Another question that could be answered using the model is what would be more beneficial for the organization to invest in, training for their current employees, or hiring new people. In the model, investing in training will be represented by increasing the efficiency of the agents from 90% to 95% and 100%. Duplicating agents in the same role describes new employees in the organization.

Table 4: Training vs New Hiring

Independent Variables	Number of Test Cases	Values Used
Make to order	-	300
Make to stock	-	0
Initial Inventory	-	200
Agents' efficiency	2	95%, 100%
Error rate	-	10%
Processing type	1	FIFO
Dependent Variables	**Number of Test Cases**	**Agent**
Average waiting time	-	Reps, planners, specs, carriers
Average time of completed orders	-	-
Total errors	-	Reworked, corrected, returned
Overall time	-	-

* The simulation will run 10 times for each test case
* This is a 2-1 experimental design case

4.2.3 Optimal Number of Individuals

From the previous experiments, we will decide which prioritization criteria should be used for this supply chain and whether investing in training or new hiring leads to high performance of the simulation. It is also important to investigate how many individuals would be required in this supply chain in order to obtain optimal results. We will conduct similar experiments to the ones in section 4.2.1, but here the focus will be on optimizing each individual role for a large organization.

Table 5: Optimal Number of Individuals in Organization

Independent Variables	**Number of Test Cases**	**Values Used**
Make to order	-	200
Make to stock	-	0
Initial Inventory	-	200
Agent of each Role (4)	5	1 - 5
Initial Efficiency	-	90%
Prioritize-type	1	Expertise or Availability
Error rate	-	10%
Processing type	-	FIFO
Dependent Variables	**Number of Test Cases**	**Agent**
Average waiting time	-	Reps, planners, specs, carriers
Average time of completed orders	-	-
Total errors	-	Reworked, corrected, returned
Overall time	-	-

* The simulation will run 10 times for each test case
* This is a 5-4-1 experimental design case

5. Results

5.1 Small Organization

For these experiments, we set the number of orders of the kind “make-to-order” from 10 to 300 to study the behavior of the system with a large volume of orders. The performance of each run is calculated by using the number of orders that were completed faster than the average completion time of all orders. From Figure 3, we can see that the performance of the simulation declines when increasing the number of orders.

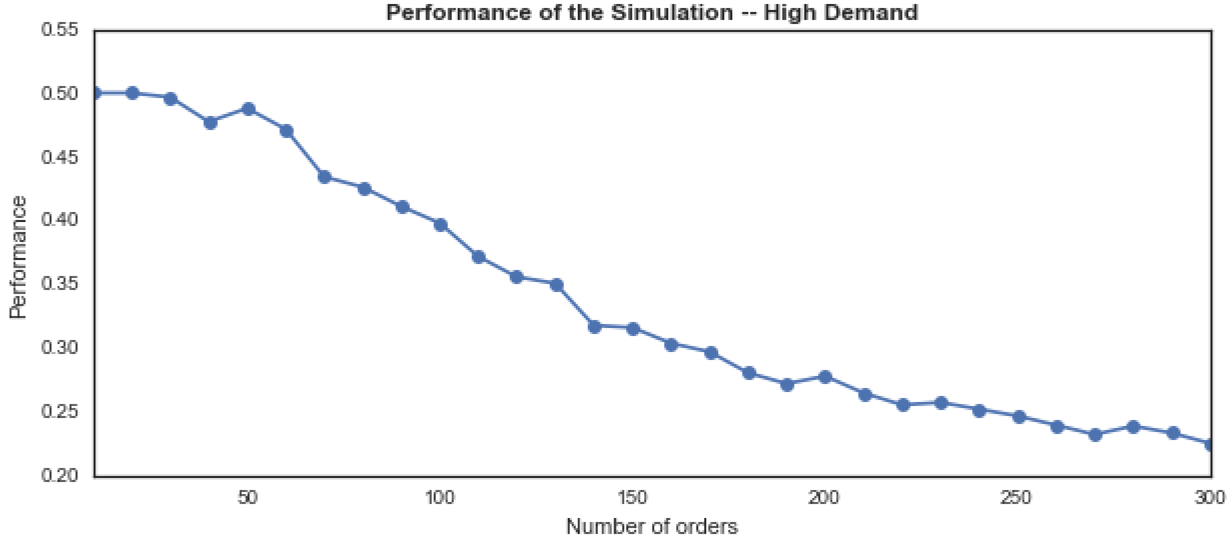


Figure 3: Performance of Simulation – High Demand

This result was expected, as the amount of errors or incomplete orders should grow in a high-demand scenario as described in Figure 4. Exceptions during the simulation run incorporate delays into the process and hence they reduce the efficiency of the model.

Figure 4: Total Errors – High Demand

In order to better understand the performance of the model, we delve into the average time and standard deviation of order completion for all the test cases. When the model is assigned between 10 - 50 orders, the completion time per order for each test case increases linearly. But once the system has to process more than 60 orders, the variability for the average time per order starts to plateau. It seems reasonable to say that after 50 orders the variability of completion times from order to order reduces due to the queue capacity of the agents, which can be as large as 40 orders. Therefore, as we increment the number of orders in the system, the waiting time of an order in the queue becomes negligible for the average completion time of each order.

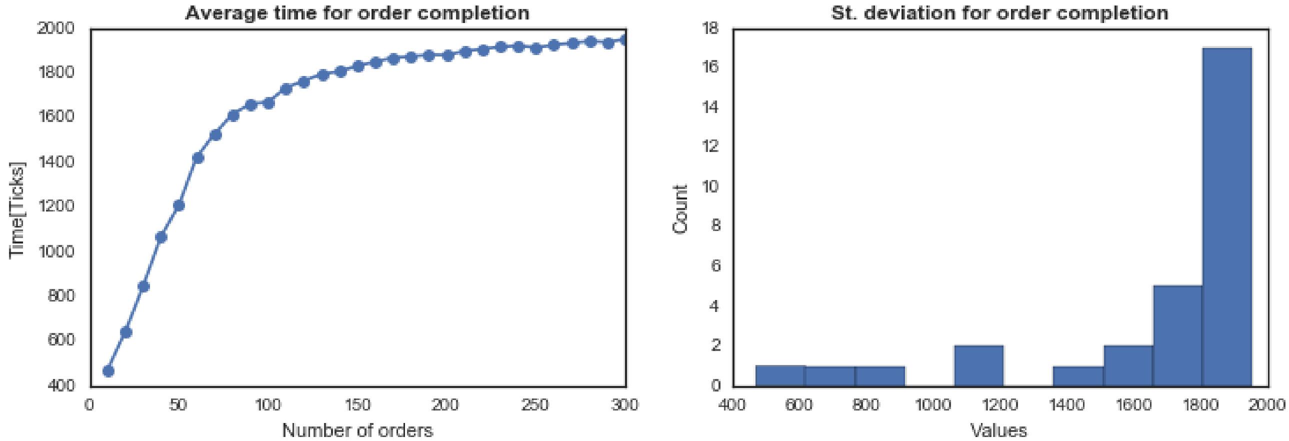


Figure 5: Average and St. deviation – High Demand

We wanted to find out which agent or variable in the supply chain had an impact on the overall performance of the simulation. We decided to analyze the processing time of the agents to determine any bottlenecks.

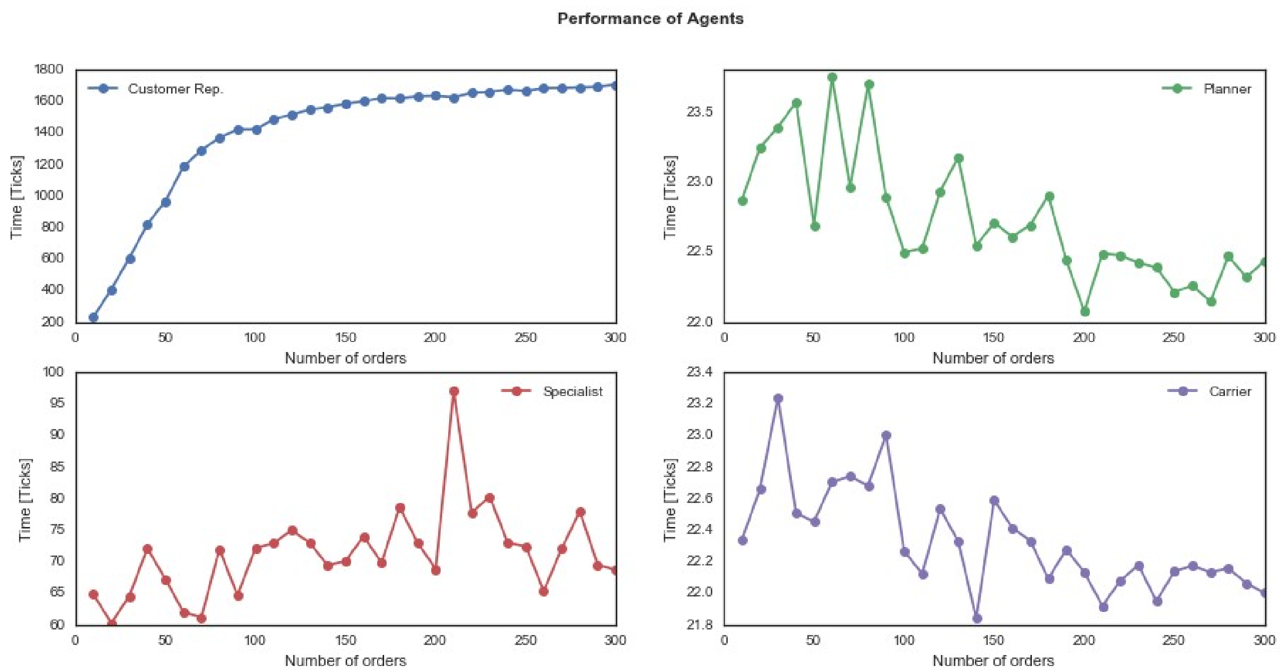


Figure 6: Performance of Agents – High Demand

From Figure 6, we can say that the customer representative is the agent that adds delays to the overall performance of the model. The processing times of the other agents are not significantly affected with a high number of orders as the customer representative is the only entity in the model that receives a backlog of orders. Therefore, it is the bottleneck of the system.

5.2 Large Organization

Now that we have found the main source of delays in the small organization, we need to identify how many agents in that role are required in order to improve the efficiency of the model. For these experiments, we give the agents a prioritization criteria to investigate what could be more efficient for an organization to focus on: expertise or availability of an agent.

5.2.1 Availability vs Expertise

When incrementing the number of customer representatives, the performance of the model for both prioritization criteria improved as described in Figure 7. It is clear that the model significantly improves its performance with two customer representatives in a high-demand scenario. However, it seems that having more than two replicates of that agent in the simulation does not contribute towards a better performing model.

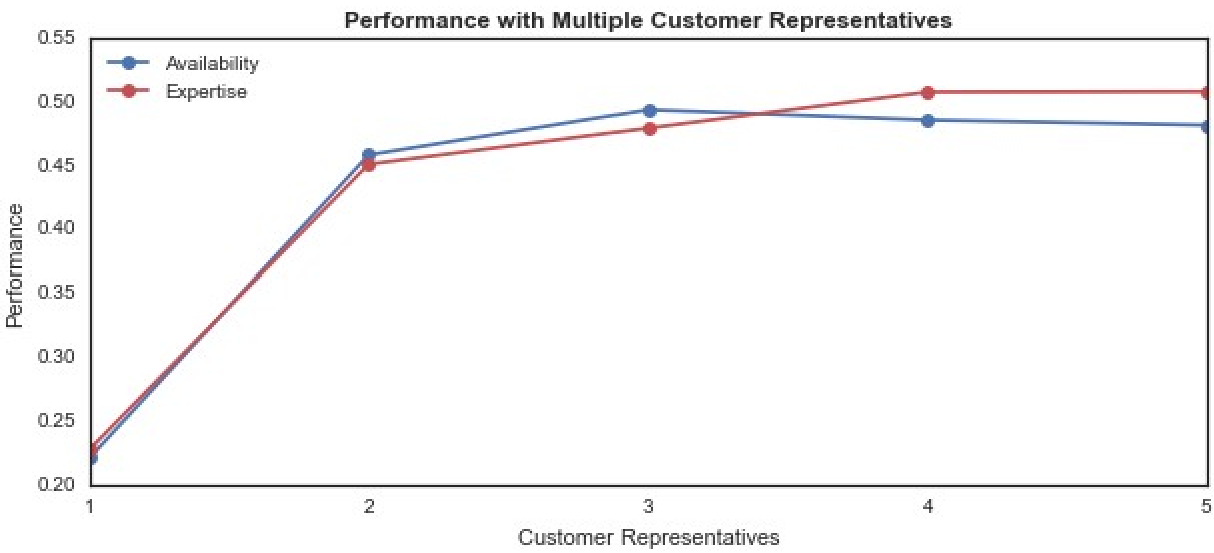


Figure 7: Performance with Multiple Customer Representatives

In order to investigate the performance of the model for both specifications, we decided to generate Figure 8 that shows the average processing time of the customer representatives as they were increased. We can see that having more customer representatives in the same role significantly reduces the amount of time that it takes them to process an order. Allowing all the available agents to process orders seems to have an almost linear decrease on their average time. Similarly, having expertise as a priority in the simulation, reduced the processing time for completing orders for the customer representatives. However, their performance starts to slow down as some of the agents in that role increased their efficiency faster and therefore, they were assigned a larger number of tasks throughout the simulation run.

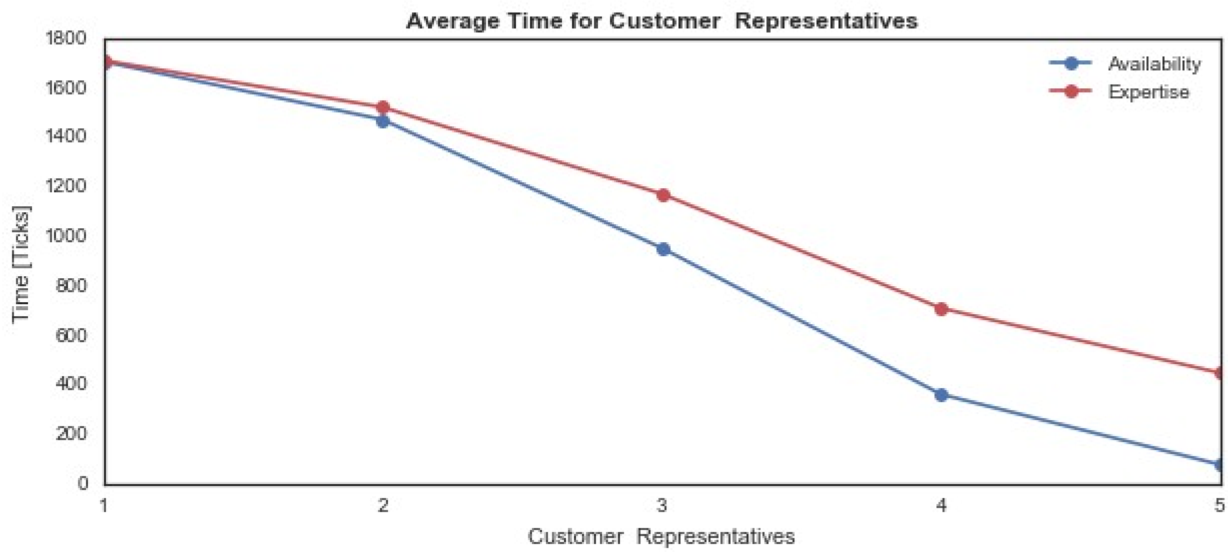


Figure 8: Average time of Customer Representatives

Because some customer representatives process a far greater number of orders when expertise is a priority, the variability in processing time from agent to agent should be drastically higher. Figure 9 describes the standard deviation in processing time across duplicated agents for each type of priority. With availability type, the amount of tasks in the system is distributed evenly across replicated agents which does not affect the variability of the model. On the other hand, having expertise as a priority type does not allow the agents in the same role to work uniformly.

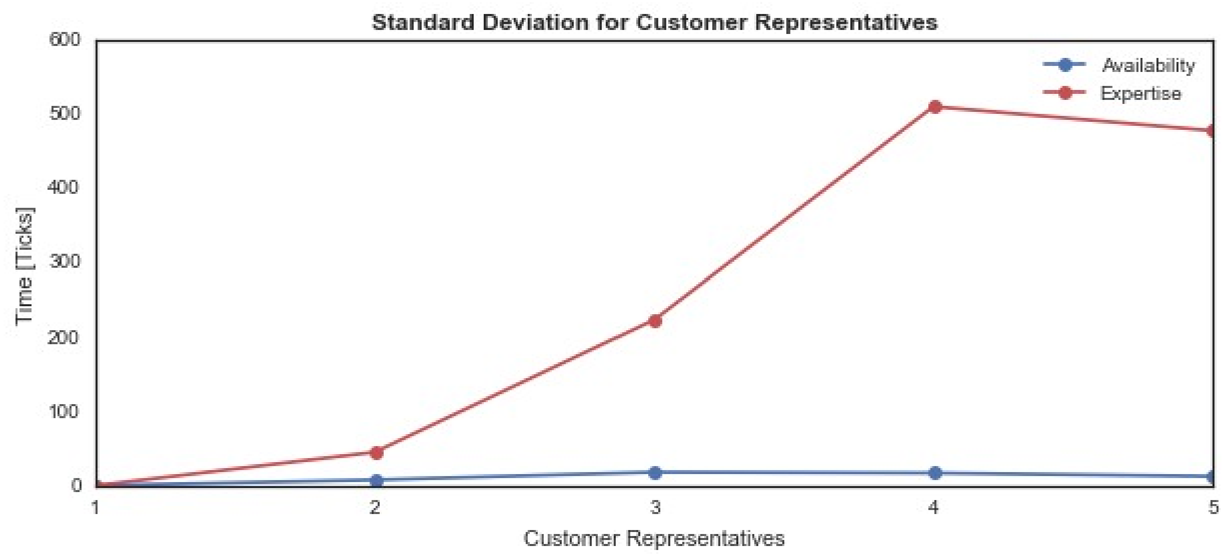


Figure 9: Standard Deviation of Customer Representatives

We were also interested in studying how the overall time to complete the orders changed when the number of customer representatives was incremented. Figure 10 shows that two agents in the system improved the completion time of the simulation for both criteria. However, incorporating more than two customer representatives to the system while keeping everything else the same did not have a significant impact on the completion time of the simulation.

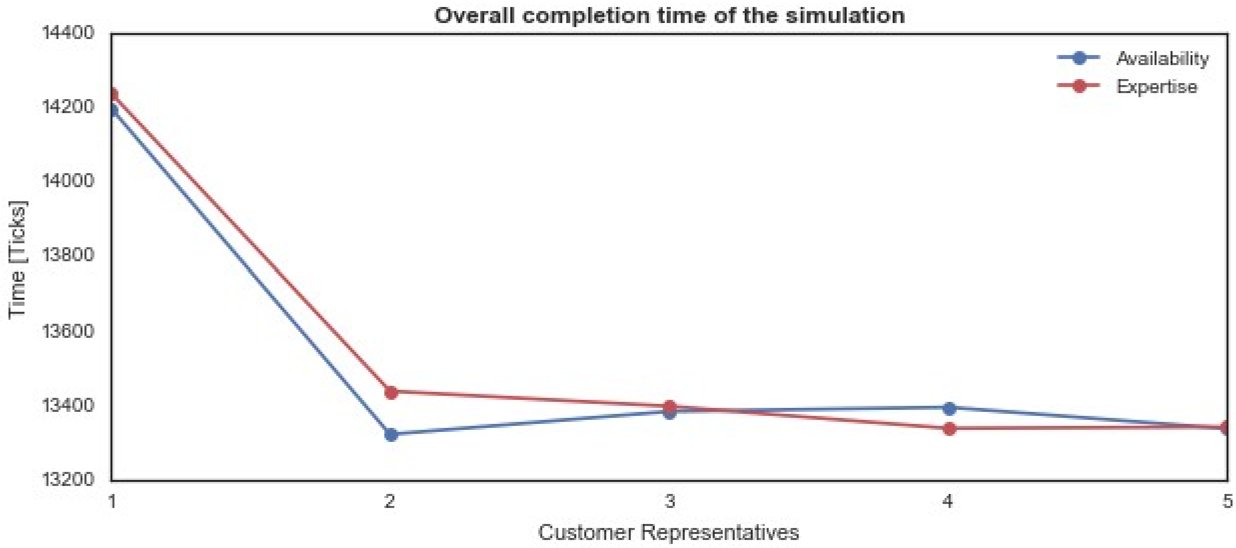


Figure 10: Overall completion time with Multiple Customer Representatives

To further our analysis, we plotted the average completion time per order for each test case. We noticed that increasing the number of agents in the bottleneck of the system, added more delays into the simulation for both types of experiments. The customer representative is the main cause of delays for the small organization, but as this agent is replicated other bottlenecks might have arisen.

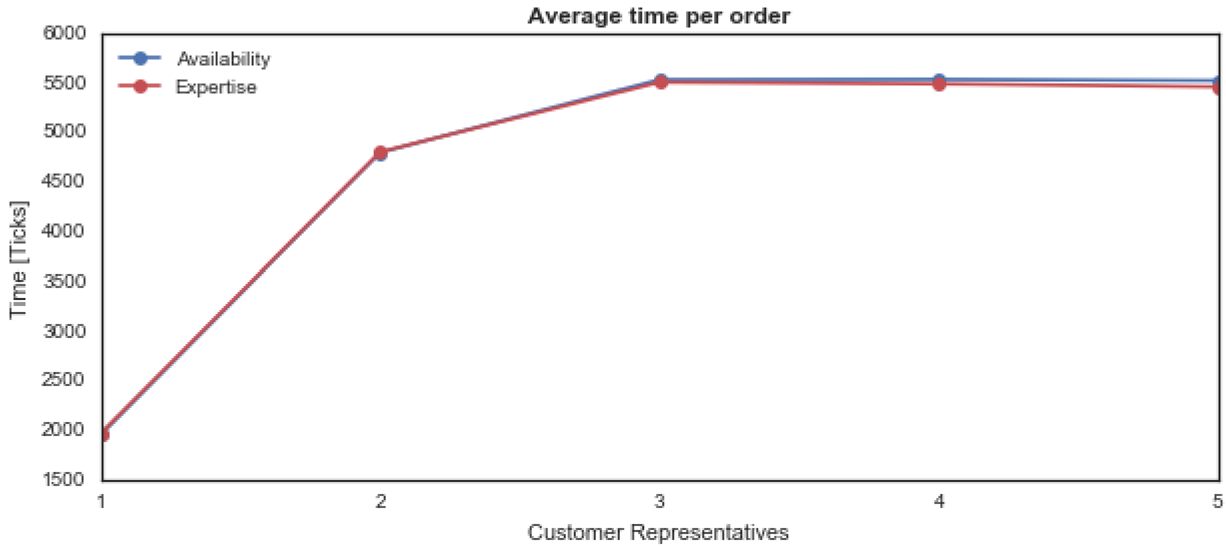


Figure 11: Average time per order with Multiple Customer Representatives

Accordingly, the average processing time for the planner, manufacturing plant, transportation specialist and carrier were analyzed to get a better understanding of our results.

From Figure 12, it is observed that the planner, manufacturing plant and specialist were clearly affected by having more customer representatives in the same role for both criteria. Since these agents were able to send orders into SAP at a faster rate, the other agents received a backlog of orders, with the exception of the carrier. We have seen consistently through Figures 7 - 11 that the variability for most features in the model becomes stagnant once the organization has between 2 - 3 customer representatives.

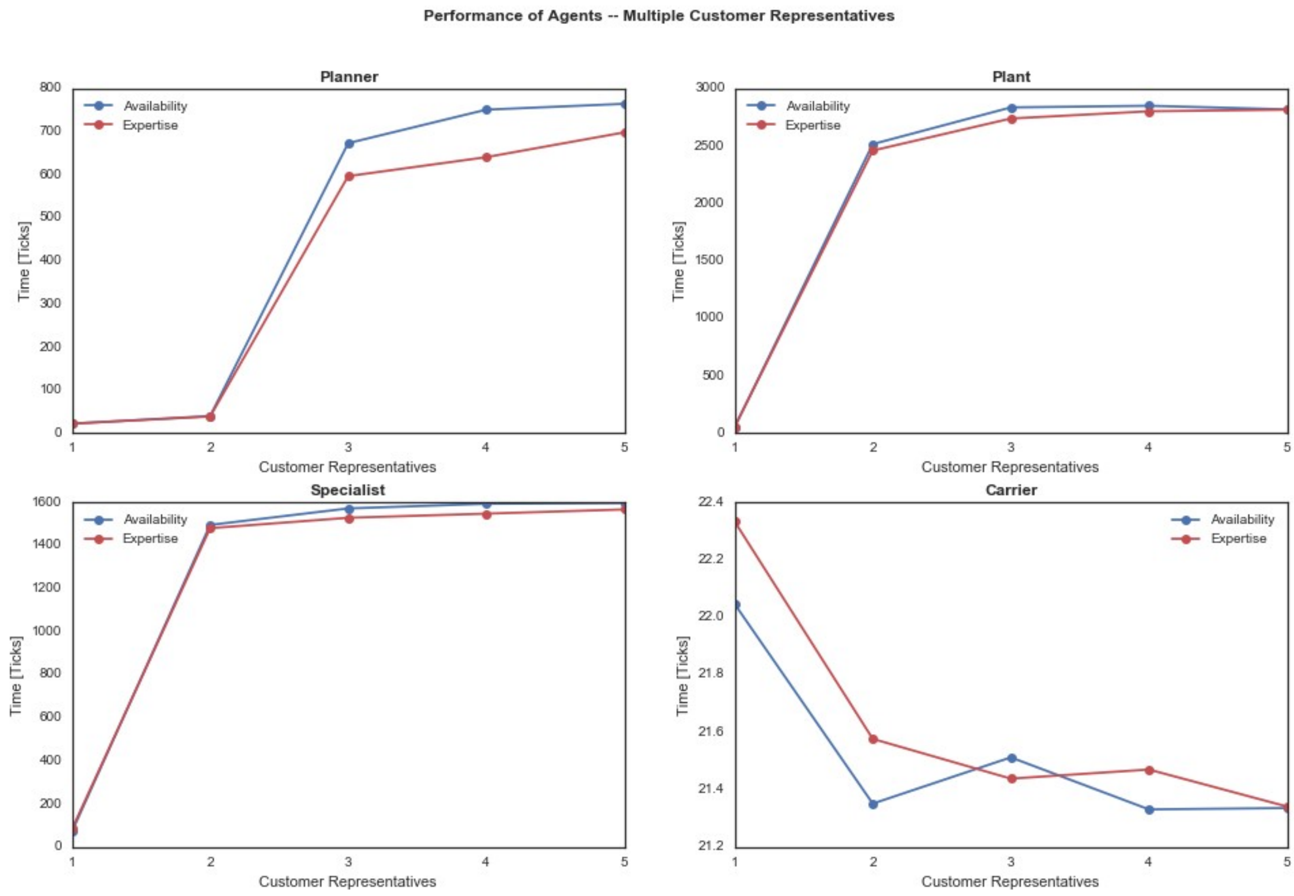


Figure 12: Performance of Agents with Multiple Customer Representatives

Our results showed that the performance of the customer representatives is more stable when the agents prioritize availability. However, an organization with an emphasis on efficiency of the overall process could obtain better results prioritizing the most expert subject on each role. This is assuming a small organization because replicating agents in each role increases the variability of their processing time. Therefore, for a large organization, prioritizing availability could lead to more consistent and predictable results overall.

5.2.2 Training vs New Hiring

In the previous section, we generated data of an organization with multiple individuals in the same role. In this section, we will use some of that data to determine if training or hiring new individuals can be more beneficial for the overall performance of an organization.

We increased the initial expertise of the agents in the small organization from 90% to 95% and 95% to 100%, and produced the following chart:

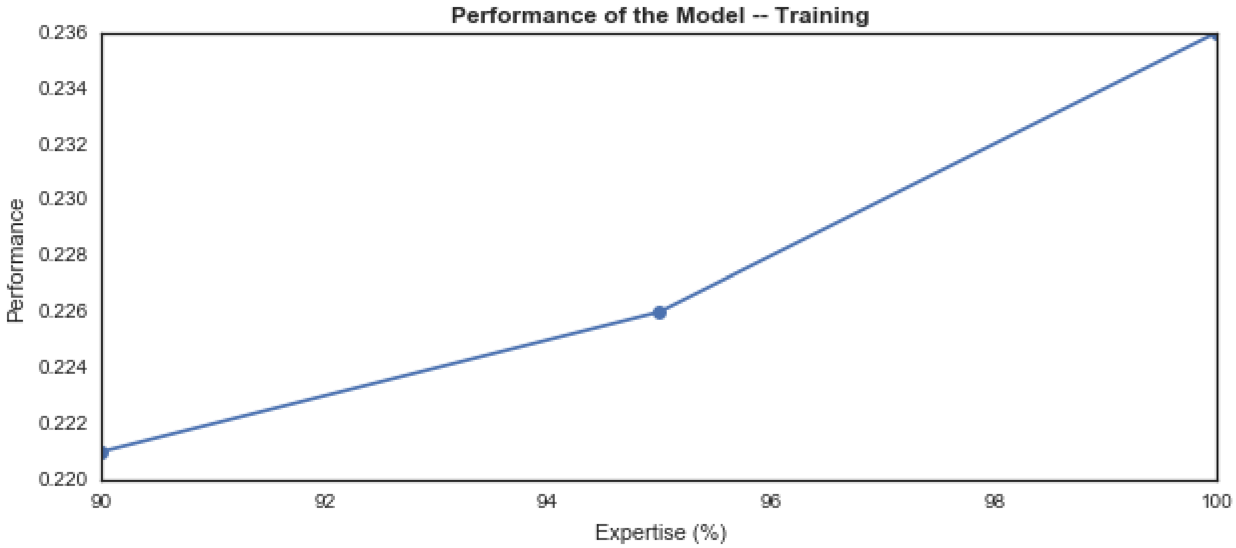


Figure 13: Performance of the Model with Higher Level of Training

From Figure 13, we can see a slight improvement on the performance of the model as the initial training of the agents was increased. In order to prove that the means of those values are significant, we performed a t-test with a significance level of 0.05 between experiments with an initial training of 90% against 95% and 100%.

Table 6: Comparison of Initial Training

Initial Training	n	Mean	SD	t-value	p-value
95%	10	0.226	0.027	-0.481	0.636
100%	10	0.236	0.019	-1.78	0.0921

The p-values in Table 6 indicate that there is not a statistically significant diff on the overall performance of the simulation when investing in training. A possible explanation for these results is that the level of expertise of the agents improves throughout the simulation run which diminishes any meaningful variability that could exist between the levels of training.

Based on the results of this section, we can conclude that hiring new individuals has a more significant effect on the overall performance of the simulation. In section 5.2.1, we saw that having two customer rep in the model was enough to obtain a beneficial improvement on the performance of the model. In addition, our results demonstrated that availability as a priority could be more beneficial for a large organization. Now, we want to determine how many people in each role are required in order to attain optimal results.

5.2.3 Optimal Number of Individuals

In section 5.2.1 Figure 12, we observed that the transportation specialist and manufacturing plant were the main point of delays when the bottleneck of the small organization was duplicated. In the following experiments, we kept two customer representatives and use availability as the priority-type since these variables were found to be optimal for the simulation. We will now increase the number of transportation specialists as we are interested in studying the influence of other individuals on the supply chain.

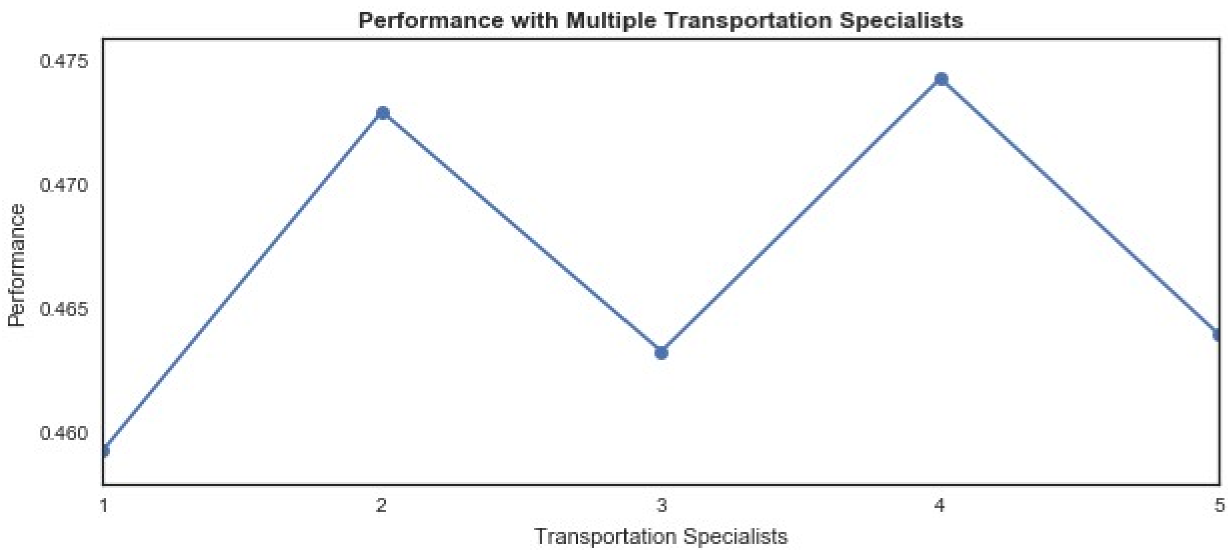


Figure 14: Performance of the Model with Multiple Transportation Specialists

In Figure 14, it is observed that the role of the transportation specialist does not directly affect the performance of the model. With 1 to 5 specialists, the performance of the model seems to fluctuate but in a very small range as shown in table 7.

Table 7: Performance of the Model with Multiple Transportation Specialists

Number of Transp. Specialists	**1**	**2**	**3**	**4**	**5**
Performance of the Model	0.459	0.473	0.463	0.474	0.464

Since performance was not improved in these experiments, we wanted to focus our analysis on another global variable of the model such as the average completion time of the orders.

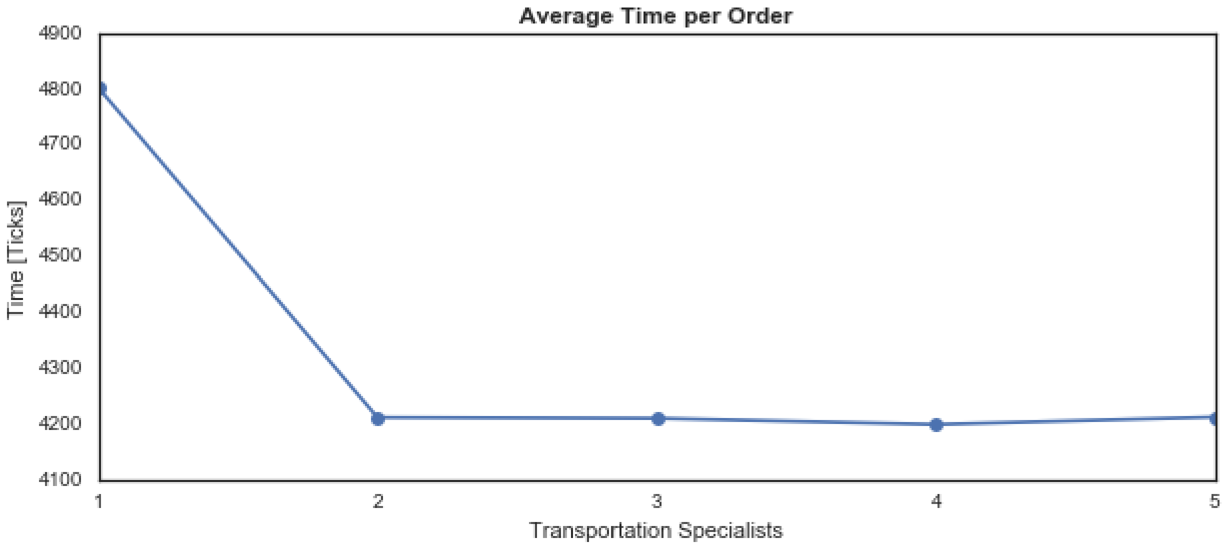


Figure 15: Average Time per Order with Multiple Transportation Specialists

Figure 15 suggests that two transportation specialists are optimal for the simulation as the average completion time per order remains stagnant after that. It was surprising to fi that even though the average time per order was reduced, the overall performance of the model did not improve. We proposed the completion times of the orders were drastically different.

From these experiments, we evaluated the processing time of the agents that have not been classified as bottlenecks of the system and generated the following Figure:

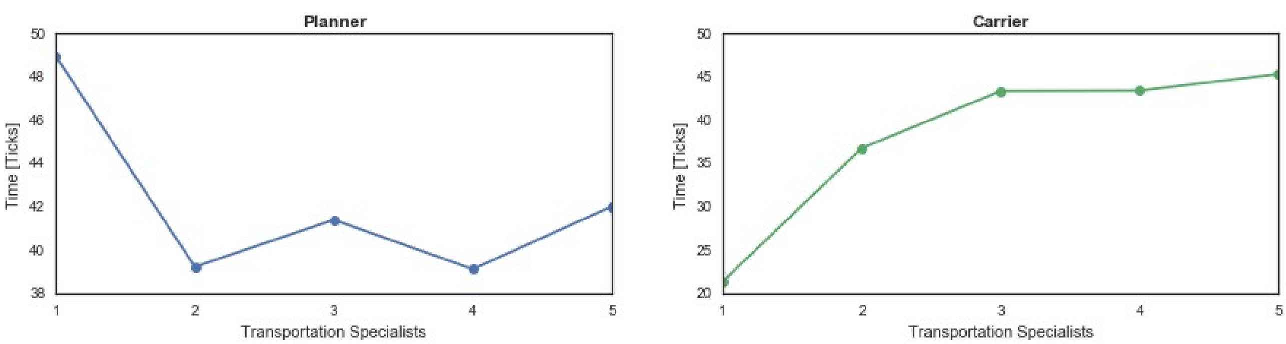


Figure 16: Performance of Agents with Multiple Transportation Specialists

The completion time of the planner decreased while two or more transportation specialists were part of the simulation, which is an unpredictable result since distribution is not connected to manufacturing. We expected delays on the processing time of the carrier as this agent was assumed to receive a backlog of orders from the transportation specialists.

Now that we have identified the carrier as another bottleneck of the simulation, we will replicate this agent to study its effect on the overall performance of the model.

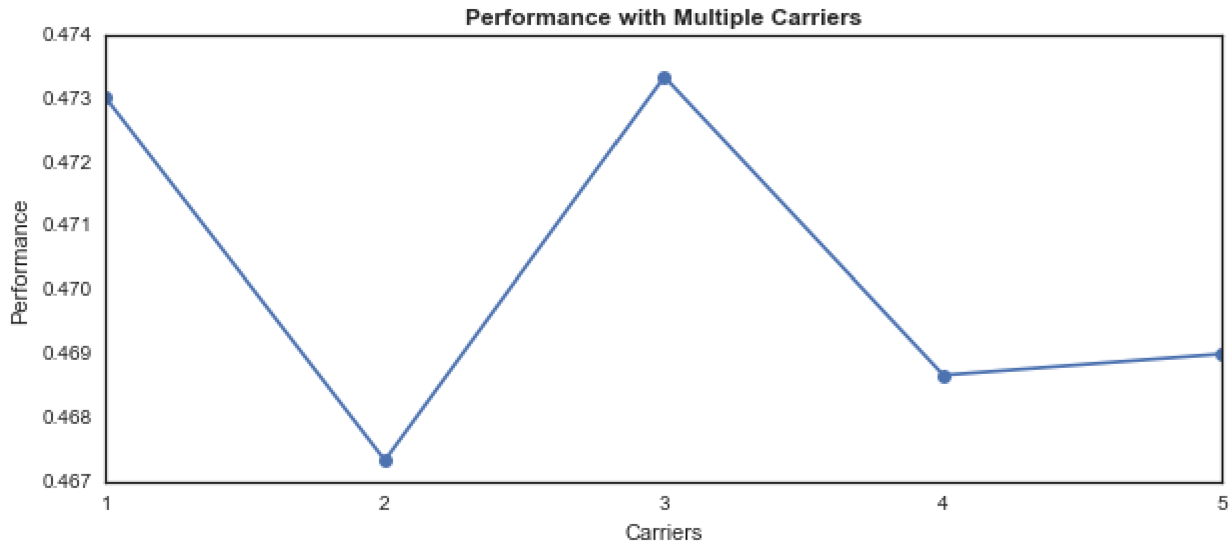


Figure 17: Performance of the Model with Multiple Carriers

From Figure 17, we can say that having a large number of carriers in the model will not contribute towards the overall performance of the simulation. Then, we consider the average time for completing the requests since this is another important variable to optimize.

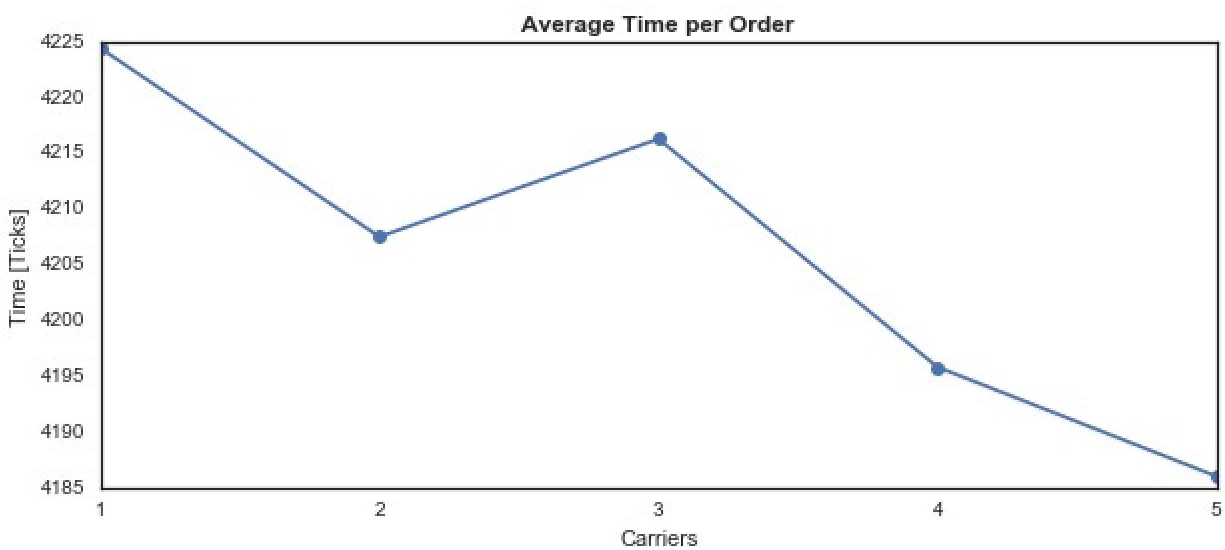


Figure 18: Average Time per Order

Figure 18 demonstrates a slight decline on the average completion time of the orders, but it is not significant enough for having more than one carrier in the model. With these results, we can conclude that the main point of delays of the model come from the customer representative and transportation specialist. Including two customer representatives improved the efficiency of the model from 22% to 47%, but it incorporated delays to the average completion time of each order. While duplicating the transportation specialist optimize the average time to complete each order by approximately 13%. However, it did not increase the efficiency of the model.

6. Conclusions and Future Work

We presented an agent-based model that simulates a simplified version of the Order-To-Cash process of a supply chain from the Dow Chemical Company. The results of our model showed that this agent-based model can capture the dynamics of a supply chain and that, it can be employed to answer organizational issues within the company. Our model was consistent with the assumption that the number of orders to be fulfilled in the simulation increases the number of errors, thereby decreasing performance. In order to improve the performance of the model, we conducted experiments for a large organization to investigate if the level of expertise of an agent matters in a specific role. Our results indicate that it is better to prioritize availability of the agents because overtime their level of expertise increases. Giving all the agents a chance to learn or get trained produces more consistent results overall.

We also conducted experiments to analyze if investing in initial training was more important than hiring new individuals as one could assume that fully trained agents would increase the performance of the simulation. However, our results suggest that agents receive training over-time, and therefore, they would not need to be fully trained before entering the organization. Hiring more people has a more meaningful effect on the overall performance of the simulation. Lastly, we determined the optimal number of individuals in each role based on the performance of the model. We noticed that two customer representatives, two transportation specialists and one agent in the other roles were sufficient to obtain better results overall. However, we theorized that the optimal number of agents for the simulation will differ from our current results, if we had the computational speed to conduct an experiment testing every possible combination of agents in each role. Due to the high computational cost of running experiments, our final goal is to move from NetLogo to either C or C++ to build a model that is large and complex enough to resemble the actual Order-To-Cash process of a supply chain.

7. References

[1] D. Emerson and S. Piramuthu, “Agent-based framework for dynamic supply chain configuration,” in *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pp. 9–pp, IEEE, 2004.

[2] S. Piramuthu, “Knowledge-based framework for automated dynamic supply chain configuration,” *European Journal of Operational Research*, vol. 165, no. 1, pp. 219–230, 2005.

[3] A. Kawa, “Simulation of dynamic supply chain configuration based on software agents and graph theory,” in *International Work-Conference on Artificial Neural Networks*, pp. 346–349, Springer, 2009.

[4] Y. Chu, F. You, and J. M. Wassick, “Hybrid method integrating agent-based modeling and heuristic tree search for scheduling of complex batch processes,” *Computers & Chemical Engineering*, vol. 60, pp. 277 – 296, 2014.

[5] Y. Chu, J. M. Wassick, and F. You, “Efficient scheduling method of complex batch processes with general network structure via agent-based modeling,” *AIChE Journal*, vol. 59, no. 8, pp. 2884–2906, 2013.

[6] Y. Jin and R. E. Levitt, “The virtual design team: A computational model of project organizations,” *Computational & Mathematical Organization Theory*, vol. 2, no. 3, pp. 171– 195, 1996.

[7] R. Cyert and J. March, *A behavioral theory of the firm*. Prentice-Hall behavioral sciences in business series, Prentice-Hall, 1963.

[8] M. D. Cohen, J. G. March, and J. P. Olsen, “A garbage can model of organizational choice,” *Administrative Science Quarterly*, vol. 17, no. 1, pp. 1–25, 1972.

[9] J. G. March, “Exploration and exploitation in organizational learning,” *Organization Science*, vol. 2, no. 1, pp. 71–87, 1991.

[10] K. M. Carley, “Organizational adaptation,” *Annals of Operations Research*, vol. 75, pp. 25–47, Jan 1997.

[11] T. Carroll and R. M. Burton, “Organizations and complexity: Searching for the edge of chaos,” *Computational & Mathematical Organization Theory*, vol. 6, pp. 319–337, Dec 2000.

[12] J. A. Effken, S. M. Gephart, B. B. Brewer, and K. M. Carley, “Using *ora, a network analysis tool, to assess the relationship of handoff to quality and safety outcomes.,” *Computers, informatics, nursing : CIN*, vol. 31 1, pp. 36–44, 2013.

[13] J. M. Swaminathan, S. F. Smith, and N. M. Sadeh, “Modeling supply chain dynamics: A multiagent approach*,” *Decision Sciences*, vol. 29, no. 3, pp. 607–632, 1998.

[14] C. Schreiber and K. M. Carley, “Validating agent interactions in construct against empirical communication networks using the calibrated grounding technique,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, pp. 208–214, Jan 2013.

[15] Z. J. Lin, X. Zhao, K. M. Ismail, and K. M. Carley, "Organizational design and restruc- turing in response to crises: Lessons from computational modeling and real-world cases," *Organization Science*, vol. 17, no. 5, pp. 598–618, 2006.

[16] J. A. Effken, B. B. Brewer, M. D. Logue, S. M. Gephart, and J. A. Verran, "Using cognitive work analysis to fi decision support tools to nurse managers' work fl w," *International Journal of Medical Informatics*, vol. 80, no. 10, pp. 698 – 707, 2011.

[17] R. M. Burton, "Computational laboratories for organization science: Questions, validity and docking," *Computational & Mathematical Organization Theory*, vol. 9, pp. 91–108, Jul 2003.

[18] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation modeling and analysis*, vol. 2. McGraw-Hill New York, 1991.

[19] U. Wilensky, "Modeling nature's emergent patterns with multi-agent languages," in *Pro- ceedings of EuroLogo*, pp. 1–6, 2001.

[20] S. Tisue and U. Wilensky, "Netlogo: Design and implementation of a multi-agent modeling environment,"

[21] M. Niazi and A. Hussain, "Agent-based tools for modeling and simulation of self-organization in peer-to-peer, ad hoc, and other complex networks," *IEEE Communications Magazine*, vol. 47, no. 3, 2009.

[22] C. M. Macal and M. J. North, "Agent-based modeling and simulation: Abms examples," in *Proceedings of the 40th Conference on Winter Simulation*, WSC '08, pp. 101–112, Winter Simulation Conference, 2008.

[23] H. Akkermans, "Emergent supply networks: System dynamics simulation of adaptive supply agents," in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pp. 11–pp, IEEE, 2001.

[24] R. García-Flores and X. Z. Wang, "A multi-agent system for chemical supply chain simu- lation and management support," *OR Spectrum*, vol. 24, pp. 343–370, Aug 2002.

[25] S. Piramuthu, "Machine learning for dynamic multi-product supply chain formation," *Ex- pert Syst. Appl.*, vol. 29, pp. 985–990, Nov. 2005.

[26] M. S. Fox, M. Barbuceanu, and R. Teigen, "Agent-oriented supply-chain management," *International Journal of Flexible Manufacturing Systems*, vol. 12, no. 2, pp. 165–188, 2000.

[27] J. C. Kunz, T. R. Christiansen, G. P. Cohen, Y. Jin, and R. E. Levitt, "The virtual design team," *Commun. ACM*, vol. 41, pp. 84–91, Nov. 1998.

Institute for Software Research • Carnegie Mellon University • 5000 Forbes Avenue • Pittsburgh, PA 15213-3890