Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds.

REDUCING INVENTORY COST FOR A MEDICAL DEVICE MANUFACTURER USING SIMULATION

Kyle Cooper
Gautam Sardar
Jeffrey Tew
Erick Wikum

Tata Consultancy Services
1000 Summit Drive
Milford, OH 41505, USA

ABSTRACT

Seeking to enter new geographic markets where expected margins are relatively tight, a manufacturer of medical devices must reduce inventory and related costs in its finished goods supply chain. The manufacturer's supply chain includes four echelons—factories, distribution centers, regional salespeople (also known as "vans"), and customers. The amount of inventory typically held and corresponding reorder policies near the customer end of this supply chain are not known. A simulation approach was selected to provide insight into those inventory levels based on assumed reorder policies. Analysis conducted using a simulation model implemented using SimPy point to significant potential savings, with the value of inventory-related savings over a four year period approaching $200 million.

1 INTRODUCTION

Emerging markets offer tantalizing potential for manufacturers to grow sales, but achieving profitable growth in these markets requires careful attention to supply chain design and execution. The growth plan for a medical device manufacturer calls for expanding into such markets. Since expected margins in these new markets are considerably tighter than in existing markets, the plan also calls for achieving significant reductions in finished goods supply chain costs. The lion's share of this manufacturer's finished goods supply chain costs are inventory-related, so reducing inventory is the company's chief objective.

While optimization is the most common approach to reduce or minimize inventory, the approach described in this paper utilizes simulation. The reason for utilizing simulation and not optimization is simple; certain data required for optimization, namely the amount of inventory typically held and reorder policies near the customer end of the finished goods supply chain is not known. Simulation provides a way to estimate that inventory and subsequently to investigate the impact on overall inventory of various policy changes. Results from the simulation are not definitive, but provide an indication of potential savings that might be achieved when implementing supply chain changes.

Not surprisingly, simulation has been applied in a variety of ways to model inventory within supply chains. For example, Gross (1969) evaluates change in inventory resulting from reduced mean lead time using simulation. Ebrahimpour (1985) uses simulation to investigate claims that Kanban systems result in less Work in Process inventory. Young (2002) proposes a simulation model that combines discrete event and continuous approaches to model the information and product flows within a supply chain. Van Der Zee (2005) describes a simulation model with flexible control structures to facilitate dialog among decision makers to improve a supply chain.

The current paper goes beyond these and other previous works in its approach to analysis, which facilitates meaningful results for the magnitude of possible inventory reduction even in the absence of key data. The remainder of the paper is organized as follows. Section 2 explains the finished goods supply chain of

978-1-4799-2076-1/13/$31.00 ©2013 IEEE

the manufacturer from factory to customer and how that supply chain operates. Section 3 provides a brief description of how a simulation model of the manufacturer's supply chain was implemented. Section 4 explains how the simulation model was verified and validated. In addition, that section describes an analysis completed using the model to identify opportunities to reduce inventory and results of that analysis. Section 5 includes concluding remarks.

2 SUPPLY CHAIN DESCRIPTION

The medical device manufacturer's finished goods supply chain includes four echelons—factories, distribution centers, regional salespeople and customers. Factories produce various types of devices and supply these devices to distribution centers (DCs). DCs supply devices to regional salespeople, also known as "vans" for the vehicles they use to deliver devices. Vans in turn supply devices to customers including hospitals and clinics. Each customer purchases a subset of device types. Each van stocks a corresponding subset of device types based on the needs of the customers it serves.

This supply chain is a "pull" system. A customer periodically orders devices from a van. In a similar manner, vans order from DCs and DCs order from factories. Figure 1 shows the structure of this supply chain, wherein orders flow from right to left, and devices flow from left to right in response to those orders. Note that each customer is served by one or more vans. In the same way, each van is served by one or more DCs, and each DC is served by one or more factories.

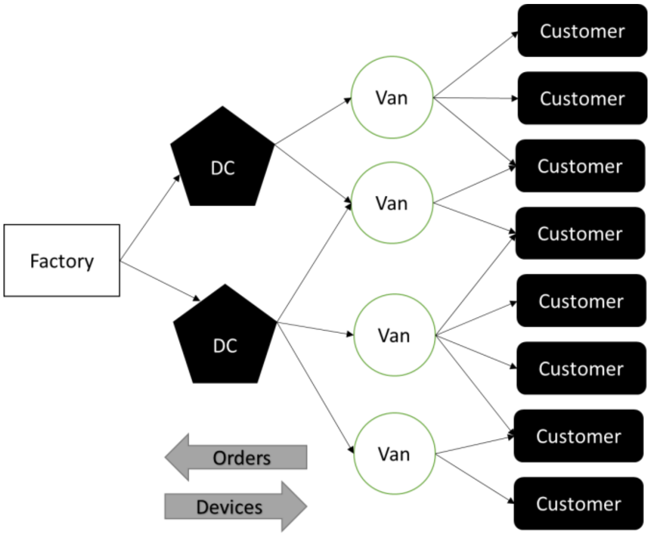

Figure 1: Structure of medical device manufacturer's finished goods supply chain

The typical amount of inventory held by each customer and individual customers' reorder policies are not known. Presumably, how frequently and how many devices a customer orders depends on that customer's current and target level of inventory and may take into account expected near-term demand.

A customer order, which specifies the number of each type of device required and a due date, can be filled in a number of ways. In most cases, the order can be filled immediately out of the inventory of a van. In case the vans serving the customer have insufficient inventory to fill the order fully and DC order lead time is short enough to meet the order due date, then some van will order the needed devices from a DC. In case no DC serving the van has sufficient inventory and factory lead time is short enough, then a DC will in turn order appropriate devices from a factory.

A stock out occurs when a customer order cannot be fully filled. Stock outs are highly undesirable for two reasons. First, a patient who needs a device may have to forego a preferred treatment option. Second, the customer may well purchase a device from a competitor, and that customer may direct future orders to that competitor as well, resulting in lost revenue.

A van may place an order to a DC not only in a reactive mode, in response to a customer order, but also in a proactive mode, to maintain on hand inventory readily available for customers. As is the case for customers, the typical amount of inventory and individual vans' reorder policies are unknown.

As mentioned earlier, a DC may place a factory order in reaction to receiving an order from a van. More commonly, a DC places a factory order proactively, to maintain inventory on hand readily available for shipment to vans. When the inventory level at a DC of a particular type of device drops below a specified reorder point, then the DC places a factory order for a number of the devices equal to a specified reorder quantity (see Figure 2). Both reorder point and reorder quantity are device specific and may vary from DC to DC.

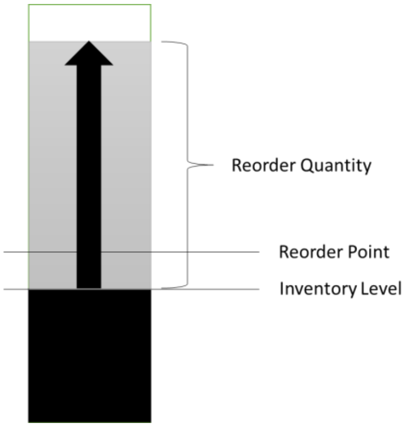

Figure 2: Form of DC reorder policy

3 IMPLEMENTATION

A simulation of the medical device manufacturer's finished goods supply chain was implemented using SimPy, an open source, discrete event simulation language for Python. The basic constructs of SimPy employed for this simulation model were processes, (resource) levels, and variables to support gathering statistics. Appendix A provides an overview of SimPy. Vignaux (2012) and Python Software Foundation (2013) provide manuals for SimPy and Python, respectively. Law (2000) provides a broad overview of simulation modeling principles.

To address the previously mentioned fact that the reorder policies for customers and vans are not known, the simulation model assumes that these policies have the same form as the DC reorder policy, with specified device-specific reorder points and reorder quantities. To address the fact that the amount of inventory typically held by customers and vans is unknown, the simulation model provides the capability to capture initial inventory levels. In the subsequent analysis conducted using the model, these initial inventory levels are set based on broad assumptions (e.g., 2 units for each device and customer or van). These initial inventory levels are adjusted up or down to avoid shortfall or decrease inventory as part of an exploratory analysis procedure described in the next section.

The primary inputs of the simulation include the following:

1. List of supply chain nodes including factories, DCs, vans and customers
2. Relationships between nodes (e.g., factory A serves DC #1)
3. Lead times between nodes in days
4. List of device types
5. Initial inventory for each node and device type
6. Reorder point and reorder quantity for each node and device type

The major processes that comprise the simulation model include the following:

1. Initialize inventory (customer, van, DC or factory)
2. Consume devices (customer)
3. Place order (customer, van or DC)
4. Process order (van, DC or factory)
5. Produce devices (factory)
6. Ship ordered devices (van, DC or factory)
7. Replenish inventory (customer, van or DC)
8. Generate inventory reports

The simulation model generates two types of inventory reports. The first logs the timing, location and amount of shortfalls. The second summarizes daily inventory levels for each node in the finished goods supply chain.

4 VERIFICATION, VALIDATION, ANALYSIS AND RESULTS

To verify and validate the model, the simulation model was run using historical customer order data for existing markets. This same data was used to conduct an analysis, the purpose of which was to gain an understanding of the magnitude of potential inventory reduction that might be achieved using the model

The purpose of verification was to ensure the simulation model was implemented correctly. The following were among the aspects of the model addressed in verification:

1. Initial inventory was set correctly for each customer, van, DC and factory.
2. Each customer order was fulfilled appropriately (out of van inventory or through a DC order) or a shortfall was logged.
3. Each van (DC) placed a DC (factory) order when inventory fell below the van's (DC's) reorder point for a specific device type and that order was filled with appropriate lead time.

The purpose of validation was to ensure the model provided a reasonable representation of the finished goods supply chain. The focus of validation was in comparing simulated to actual DC inventory.

Once verified and validated, the model was employed within the following heuristic procedure to eliminate incidents of shortfall and to reduce inventory requirements within the finished goods supply chain. The heuristic avoids shortfalls, which are highly undesirable. In addition, the heuristic recognizes that inventory policy parameters including reorder point, reorder quantity, and initial inventory can be adjusted downward to reduce time-weighted inventory across the supply chain network. In the description of the heuristic shown in Figure 3, the term *inventory* refers to a particular device type stocked at a customer or van node.

Suppose that, when the simulation model is run, an inventory shortfall of two units occurs for some device and customer. One way to eliminate that shortfall is to increment initial inventory for that device and customer by two units. To avoid holding too much inventory over time, that method is only used if the shortfall occurs at the start of the simulation. Otherwise, the shortfall is eliminated by increasing the reorder point by two. As a result, the inventory level after the last reorder prior to the shortfall will be two higher, thereby covering the shortfall.

Now, suppose that simulation results show that the minimum inventory level is three for some device and customer with reorder point ten and reorder quantity five. Reducing the reorder point to seven will result three less units in inventory at the time of the first reorder and thereafter. As a result, the minimum inventory level will be reduced to zero and the average inventory level will be reduced by nearly three units (depending on the portion of the time horizon represented by the first reorder cycle).

Results of running the heuristic procedure using historical customer order data suggest the potential to achieve significant benefits for existing markets (and subsequently for developing markets). Over a four year period, the value of reduced inventory was nearly $200 million. The reduced inventory was accompanied by an improvement in inventory turns of nearly 70%. Because these results are based on assumptions regarding the typical amount of inventory held by and reorder policies of customers and vans, then the actual reduction in inventory achieved during implementation would likely be less, but would still be quite significant.

```
Repeat
    Run the simulation model

    Eliminate inventory shortfalls:
        For each inventory shortfall incident:
            If shortfall occurs at start of time horizon, then
                Increment initial inventory for the device by the minimal
                amount required to eliminate the shortfall
            Else
                Increase the reorder point for the device by the minimal
                amount required to eliminate the shortfall
            End If

            If the level of inventory over the simulation time horizon is consistently below the reorder point, then increase the reorder
            quantity as appropriate
    Reduce inventory over horizon:
        For each inventory with no shortfall incidents:
            Reduce reorder point, reorder quantity, and initial inventory in
            turn by maximal amount such that no shortfall results

Until no inventory shortfall incidents and no inventory reduction opportunities exist
```

Figure 3: Shortfall Elimination and Inventory Reduction Heuristic

5 CONCLUDING REMARKS

This paper demonstrates that simulation can play a useful role in inventory management when data required to support the most common approach, optimization, is not available. In the case of a medical device manufacturer that seeks to enter emerging markets, the amount of inventory typically held and reorder policies at the customer end of the finished goods supply chain are unknown. Simulation facilitates exploratory analysis to understand how reducing initial inventory, reorder points and reorder quantities can impact supply chain inventory. Analysis conducted using a simulation model suggests that significant reductions in inventory for existing markets are possible. Based on this analysis, the manufacturer can implement changes to reduce inventory in existing markets and can leverage learnings, and the simulation model, to achieve profitable growth in new markets.

Given that the manufacturer's finished goods supply chain is a pull system and that the demand (customer) end of this supply chain is somewhat of a mystery, some obvious questions arise. Is it technically and politically feasible to collect inventory level and reorder policy (or order forecast) data from customers and/or vans? What analytic approach would be best to utilize for the supply chain if some or all of that data were available? The sizable potential benefits suggested by the analysis described above suggest that further investigation is certainly warranted.

A SIMPY OVERVIEW

SimPy is a discrete-event simulation framework for the Python programming language. It has been continuously developed and refined since 2003. SimPy is open source software with usage governed by the

Lesser Gnu Public License. It has evolved to provide two primary object types to simulation programmers—active entities and passive entities. Active entities have behaviors defined by the programmer. Passive entities are congestion points in a model and retain certain behaviors even in drastically different simulations.

Active entities in SimPy are programmed as SimPy processes. Processes are Python classes with a user-defined simulation behavior. In this simulation, one process type is the shipment. A shipment process is created each time an order is received at the factory, delivery center, or van. The process utilizes resources and passive entities, such as inventories, to complete a shipment. The entire functionality of a process must be defined in the process class definition. Particularly, the process execution method must be defined in the class definition. The process execution method is a function that is called when the object is activated by the simulation. It contains one or more yield statements which is required for SimPy to generate events.

Three types of passive entities are provided by SimPy—resource, level, and store. Resources have some number of units which can be used by a process. The simulation described herein uses resources to model dock doors at each warehouse. A truck process can request a dock door if one is available. While the truck is using the dock, the dock is unavailable to other trucks. Levels model the supply of homogeneous material. A process can add or withdraw material from a level and the amount of supply in the level is modeled by a single number. Packing boxes are a material modeled with a level. A store models a collection of individual items. The items can be any Python object including SimPy processes, such as trucks.

A major component of many supply chain simulations is shipping. A SimPy process may be created to handle all shipments in the simulation. The following code implements a simple shipping module.

```
1 class Shipment(Process):
2     def run(self, src_inv, dest_inv, time, amt):
3         yield get,self,src_inv,amt
4         yield hold,self,time
5         yield put,self,dest_inv,amt
```

This shipment process simulates a shipment by extracting items from an inventory, holding them for some amount of time, and then depositing the items into a different inventory. The first line defines the class, giving it a name, and inherits the SimPy process class, which is required to create a SimPy process. The second line defines a function with the `def` keyword and gives it a name and parameters. All class functions must include a reference to the object as a parameter. This is the `self` keyword. The other parameters are required to perform the shipment: (1) the inventory from which product is being taken, (2) the inventory into which product will be deposited, (3) the amount of time the product will be in transit and (4) the amount, quantity, of product being shipped. No product type is specified. That may indicate that inventories hold only one type of product or that the items in the inventory were prepared elsewhere in the simulation. Since the function body contains yield statements, this function, called `run`, is a valid process execution method in SimPy.

REFERENCES

Ebrahimpour, M and B.M. Fathi. 1985. “Dynamic Simulation of a Kanban Production Inventory System.” *International Journal of Operations & Production Management*. Bradford: 5 (1) 5-14.

Gross, D. and A. Soriano. 1969. “The Effect of Reducing Lead Time on Inventory Levels – Simulation Analysis.” *Management Science* 16 (2) (October): B-61-B-76.

Law, A. and D. Kelton. 2000. *Simulation Modeling and Analysis*. 3rd Ed., McGraw-Hill Inc.

Python Software Foundation. 2013. *Python Language Reference*. http://docs.python.org/3/reference/index.html#reference-index.

Van Der Zee, D.J. and J.G.A.F. Van Der Vorst. 2005. “A Modeling Framework for Supply Chain Simulation: Opportunities for Improved Decision Making.” *Decision Sciences*. 36 (1) (February) 65-95.

Vignaux, T., K. Muller and B. Helmbold. 2012. SimPy Manual. http://simpy.sourceforge.net/SimPy_Manual/Manuals/Manual.html.

Young, H.L, K.C. Min, J.K. Seo, and B.K. Yun. 2002. “Supply Chain Simulation with Discrete-Continuous Combined Modeling.” *Computers and Industrial Engineering*, 43 (1-2) 375-392.

ABOUT THE AUTHORS

KYLE COOPER is a Researcher in Tata Consultancy Services' Cincinnati Innovation Lab. Kyle earned his B.S. in Computer Science from the University of Louisville. His interests include supply chain design and analysis using simulation. His email address is kyle.cooper@tcs.com.

GAUTAM SARDAR is Chief Scientist in Tata Consultancy Services' Supply Chain Center of Excellence. Gautam earned his Ph.D. in Systems and Control Engineering from Indian Institute of Technology, Bombay. His research interests include manufacturing/supply chain risk modeling, control theoretic frameworks for supply chain simulation/visualization and complex event processing. Gautam is recipient of "Pros to Know Award by Supply & Demand Chain Executive" in 2013 and "TCS Innovator Award" in 2006. Gautam's email address is gautam.sardar@tcs.com.

JEFFREY TEW heads Tata Consultancy Services' Cincinnati Innovation Lab as Chief Scientist. Dr. Tew earned his Ph.D in Operations Research from Purdue University. Dr. Tew's research interests include transportation, supply chain strategy, business development and technology management. His email address is jeffrey.tew@tcs.com.

ERICK WIKUM is a Principal Scientist in Tata Consultancy Services' Cincinnati Innovation Lab. He earned a Ph.D. in Operations Research from the Georgia Institute of Technology. Dr. Wikum has applied analytics to business problems across numerous industries with special emphasis in freight transportation. Erick's email address is erick.wikum@tcs.com.