

SIMULATING SKU PROLIFERATION IN A HEALTH CARE SUPPLY CHAIN

Manuel D. Rossetti

Yanchao Liu

Dept. of Industrial Engineering
4207 Bell Engineering Center
Fayetteville, AR 72701, USA

Dept. of Industrial and Systems Engineering
3270 Mechanical Engineering
1513 University Ave, Madison, WI 53706, USA

ABSTRACT

This paper investigates stock keeping unit (SKU) proliferation and its impact on a health care supply chain. There are two types of proliferation: acceptance and adoption. The acceptance case occurs when a hospital requests a SKU and the SKU is not carried by the system. The adoption case occurs when the SKU is not carried by the requesting hospital, but is carried by some other hospital within the system. A multi-item, multi-echelon supply chain model with load building and delivery details was constructed to examine cost and performance tradeoffs. Object-oriented modeling and implementation techniques for the SKU proliferation simulation are described. The results indicate that the acceptance and adoption jointly affect the performance of the supply chain in terms of the service level (demand fill rate) and the cost. The models and results provide a test bed for addressing and mitigating the effect of proliferation within healthcare supply chains.

1 INTRODUCTION

A healthcare supply chain is a typical multi-echelon, multi-item inventory system. It usually consists of four layers of players, they are (1) external supplier layer, which includes manufacturers, wholesalers, vendors, sales representatives of medical, surgical or pharmaceutical supplies; and there might be thousands of supplier entities in this layer; (2) a consolidated service center (CSC), which acts as a group purchasing organization (GPO). CSC purchases the supplies directly from various external suppliers and stocks them in its warehouse; (3) strategic service unit (SSU) layer, which includes hospitals, clinics and pharmacies, etc. SSUs directly meet the end customer's demand; and (4) end customer layer, which include physicians, patients, etc.

A distinguishing feature of the healthcare supply chain is that it is a customer driven system and what is more, the customers may demand virtually any item, be it already in the system or not in the system. For example, in an operating room, different physicians may prefer different brands (from different suppliers) of functionally equivalent devices; a same physician may suggest different brands of implants to different patients. Because of direct-to-patient advertisement, patients now have more voice in choosing medical devices used on them. Therefore, hospitals have to incorporate in their procurement and inventory systems more and more different types of SKUs that have the same functionality. This is a major cause of SKU proliferation in healthcare systems.

In a multi-echelon, multi-item supply chain environment, the situation becomes more complex. As a CSC serves several SSUs, a new item request at a SSU may not be new to the CSC; instead, it may drive the CSC to adjust its inventory. On the other hand, if the new item request at a SSU is also new to the CSC, this indicates that the item is new to the whole system and the CSC must set up inventory to accommodate the item. Apart from this, the shipment schedules between CSCs to SSUs could also be complicated by SKU proliferation.

This paper examines the effect of SKU proliferation on the costs and performance of a simplified health care supply chain. The remainder of this paper is divided into four sections. In Section 2, the SKU proliferation model as well as the cost model is presented. Section 3 briefly describes the model implementation techniques. Section 4 discusses the experimentation and results, and conclusions are given in Section 5.

2 SIMULATION MODELING

2.1 SKU Proliferation Modeling

We model the SKU proliferation process via two cases: new SKU acceptance and existing SKU adoption. In the acceptance case, the new SKU demanded at an SSU is also new to the CSC, therefore, both the SSU and the CSC must set up inventory for the new SKU; while in the adoption case, the new SKU demanded by a SSU already exists in the system i.e. in the CSC, and only the SSU needs to set up inventory for the new SKU. A brief description of the proliferation concepts and procedures follows based on the following notation:

N	The number of item categories in the system
M	The number of SSU locations in the system
$\xi = \{C_1, C_2, \dots, C_N\}$	The set of item categories
$K = \{k_1, k_2, \dots, k_N\}$	The set of the numbers of item types in category 1 to N
$\omega_{i \in \{1, 2, \dots, N\}} = \{I_{i,1}, I_{i,2}, \dots, I_{i,k_i}\}_{k_i \in K}$	The set of item types in category C_i
$\varepsilon = \{L_1, L_2, \dots, L_M\}$	The set of SSU locations

The CSC maintains a list of all item types currently in the supply chain, represented as the set A , defined as:

$$A = \bigcup_{i=1}^N \omega_i$$

Each SSU j maintains a list of the item types it accommodates. Note that an SSU may have only a few but not all item types in a category, define

$$B_j = \bigcup_{i=1}^N v_{i,j} \subseteq \omega_i \quad \forall j = 1, \dots, M$$

as the set of SKU's in SSU j , where $v_{i,j}$ denotes the set of item types in category i at SSU j . Each item type $I \in A$ maintains a list of SSU locations that accommodate it, define $D_I \subseteq \varepsilon \quad \forall I \in A$.

The model keeps track of the set of SKUs currently in the system, i.e. A and the sets of SKUs in each SSU j , i.e. B_j . When a proliferation event occurs at a SSU, say SSU j , the model will check if the demanded item type is already in A . If it is not in A , the model will know that a new item type has to be added to the system so it will turn to the Acceptance procedure. If the proliferating item type is in A , then the model goes on to check if the SKU is in B_j , if not, the model will know that this item type will diffuse to a new SSU location so the Adoption procedure should be called.

We let the model initiate the proliferation incidents actively. The proliferations (both Acceptance and Adoption) occur according to certain rates that are related to the type of the proliferation, or other attributes such as the item category and SSU location associated with the proliferation. During such an event, the model first processes the proliferation by updating its carefully managed sets and lists, then according to the updated sets and lists, realizes the physical proliferation, such as adding inventory to the appropriate locations, generating demands for the proliferated item types and making necessary adjustments in the supply chain hierarchy, etc.

The model has to decide two things, namely, when to proliferate and what to proliferate. These two aspects are closely related to each other and together they uniquely determine a proliferation incident. As mentioned before, in this model the inter-proliferation times (when to proliferate) are stochastic. Therefore, the proliferation specifics (what to proliferate) should be given (or known) when the proliferation occurs. For example, we may describe a SKU proliferation incident as “approximately every 2 months, a new item type (e.g. a brand of cardiac stent) will be added to item category 1 (e.g. category of the cardiac stent) at SSU 1 (e.g. Mercy hospital at Fort Smith, AR)”. In this scenario, the proliferation context “category 1” and “SSU 1” are clearly given, while the “every 2 months” describes the proliferation times, which are random. The two cases as well as their respective pseudo-code are as follows:

Acceptance Case

In the Acceptance case, the target item category and SSU location are given as parameters, denoted by C_i and L_j respectively. The first step is to create the new item type I_{new} to be added. I_{new} is created according to the attributes of C_i , so that it has statistically similar characteristics (such as price, cube and supplier lead time, etc) with other item types in category C_i . After I_{new} is created, it is added to the set ω_i that holds all item types in category C_i , it is also added to the set $v_{i,j}$ which holds all item types of category C_i at location L_j . Then a set $D_{I_{new}}$ is created and attached to the new item type I_{new} with L_j being its first member. Note that the set $D_I, I \in A$ keeps track of all SSU locations that accommodate the item type I , which is useful information for the Adoption case. Lastly, the sets A and B_j are updated.

Acceptance $|C_i \subseteq \xi, L_j \subseteq \varepsilon$

Begin
 Generate a new item type I_{new} for the category C_i
 Add I_{new} to the set ω_i , update $k_i = k_i + 1$ in the set K .
 Add I_{new} to the set $v_{i,j}$.
 Initiate $D_{I_{new}} = \phi$, add L_j to $D_{I_{new}}$.
 Update A and $B_j, \forall j = 1, \dots, M$.
 End

Adoption Case

In the case of adoption, the item type that is to be diffused (and adopted by other SSU locations) is given as a parameter, denoted by $I_{i,j}$. Note that $I_{i,j}$ indicates the j -th item type in category i . The procedure begins by checking if the given item type $I_{i,j}$ has any locations to diffuse to or not. This is done by comparing the set $D_{I_{i,j}}$ that contains all the locations currently accommodating item type $I_{i,j}$ with the set ε that contains all the locations in the system. If the two sets have the same size, then this means the item type $I_{i,j}$ cannot be adopted by any location because all locations already have it; otherwise, the difference between the two sets is just the set of candidate locations to which $I_{i,j}$ can be adopted. Therefore, the model proceeds to select the target location L_s from the set $\varepsilon - D_{I_{i,j}}$. Note that the selection criteria could vary to account for different considerations. In this paper, random selection is used. After L_s is selected, the connection between the item type $I_{i,j}$ and the location L_s must be established. This is done by adding $I_{i,j}$ to L_s 's list B_s , and adding L_s to $I_{i,j}$'s list $D_{I_{i,j}}$.

Adoption $|I_{i \in \{1,2,\dots,N\}, j \in \{1,2,\dots,k_i\}}$

Begin
 If $|D_{I_{i,j}}| = |\varepsilon|$, goto End.
 Select* an L_s from the set $\varepsilon - D_{I_{i,j}}$
 Add $I_{i,j}$ to the set B_s .
 Add L_s to the set $D_{I_{i,j}}$.
 End

These procedures make sure that the SKU proliferation proceeds orderly and organized over the whole time horizon. During the physical proliferation, the inventory for the new item type will be added to appropriate locations and it will share the inventory quota of the category. In this paper, we assume that all the item types within a category evenly split the demand process of that category, as well as evenly share the inventory quota of that category. For example, if the demand rate for $v_{i,j}$ is r and the inventory quota is q , then the demand rate and the inventory for each item type $I \in v_{i,j}$ are $r/|v_{i,j}|$ and $q/v_{i,j}$, respectively.

2.2 Cost Modeling

Capturing the cost is an essential objective of the supply chain modeling. In the system context under study, the total operating cost can be classified into four broad categories.

1. **Procurement costs.** Procurement costs are those associated with the acquisition of items at the CSC. They include a reorder cost and a transportation cost. A reorder cost is charged whenever an order is issued. The transportation cost is proportional to the number of items being transported in one shipment package. Therefore the annual transportation cost from suppliers to the CSC for all SKUs can be easily calculated, given the cost rate and the inventory policy.
2. **Inventory holding costs.** Inventory holding costs are incurred when items are stored for a period of time. They include a capital cost and warehousing costs. For modeling simplicity, we assume the annual cost for keeping one unit of the same SKU to be constant, both at the SSUs and at the CSC.
3. **Transportation costs.** The transportation costs involve the costs of shipments from the CSC to the SSUs. They include the load building labor cost, the vehicle maintenance cost, drivers' wages and fuel cost.

4. **Shortage costs.** Shortage costs are paid when customer orders are not met. In healthcare industry, shortage is least welcomed because of the public's high expectations on the service quality. In this system context, we only account for direct-to-customer shortages that happen at SSUs and omit the those at the CSC, to avoid a duplicate penalty.

2.3 Transportation Modeling

In this system context, we are concerned with the transportation of items from the CSC to SSUs. Since SSUs usually keep low inventories as safety stock, the replenishment cycle is very short. In other words, transportation between CSC and SSUs happens frequently. The SSUs will send orders to the CSC in every replenishment cycle. The CSC, which constantly receives orders from many SSUs, will put the orders in a buffer until a shippable load can be built, then it will dispatch a carrier for shipment. Depending on the size of the load, the CSC may have several options in choosing a suitable type of carrier. Three load building rules are modeled:

1. **By quantity.** A load is formed when the quantity of orders accumulated in the buffer surpasses the minimum quantity for an economic shipment. Here quantity can denote the number, weight or cube of the ordered items, or any combination of limits for these attributes. For example, a load building rule may require that either the total weight in the load should exceed a certain minimum level or the total cube should exceed a certain level. Load building by quantity could ensure high utilizations of carriers and the economy of transportation.
2. **By Schedule.** The CSC keeps a fixed schedule for shipments to SSUs. Whenever a scheduled shipment time is reached, a load is built out of whatever is in the buffer and the load is then shipped immediately. If the load happens to be very small, the CSC will dispatch a small-sized vehicle; otherwise, it will dispatch a large-sized vehicle. If the order quantity is moderate, load building by schedule can guarantee a targeted fill rate. However, by the time of scheduled shipment, if the accumulated order quantity is so big that one truck load is not enough to hold all the orders, some orders will have to wait in the buffer for the next shipment and may not be able to reach the SSU on time.
3. **Mixed rule.** Duplicate rules can be imposed on the load building to make the shipment both flexible and reliable. For example, the CSC could keep fixed shipment schedules but if the buffer grows sizable enough to form an economic load, the CSC could form the load and make an extra shipment before the scheduled shipment occurs. In this way, both the fill rate and truck utilization level can be kept high.

After a load is built, shipment transportation starts immediately. The carrier takes the load to the destination and returns empty. Two types of carriers are modeled, trucks and trailers. Trucks have smaller capacity but consume less fuel than trailers for the same traveling distance. The load building fees and driver wages are also lower than those of a trailer.

3 MODEL DEVELOPMENT AND IMPLEMENTATION

In this research, the model is conceptualized and developed following object-oriented principles and is implemented using the Java programming language. The program is built upon the JSL. For more information about the JSL, readers are referred to Rossetti's homepage at www.uark.edu/~rossetti/. Rossetti (2008) developed an object-oriented framework for simulation modeling. The framework was implemented using Java and has grown into a full-fledged simulation tool – Java Simulation Library (JSL). The JSL can facilitate the modeling of discrete-event systems in Java using the event and process world-views with full support for random variable generation and statistical output collection. It distinguishes itself from other similar libraries by its flexibility and extendibility under an open-source paradigm. For example, the reliance on observer-based collection of statistics is unique to the JSL and provides great flexibility. In addition, the JSL can readily use different random number generators, different event scheduling calendars and different statistical collection techniques, etc., which facilitate research of these topics within a consistent platform (Rossetti 2008).

Figure 1 provides an overview of the model structure. It is neither a direct translation from the implementing class structure, nor a complete diagram explaining any implementing details. Rather, it reflects the general guideline underlying the overall model development and implementation process. In the figure, solid lines represent the reference relationship and the dashed lines represent message exchanging channels between different modules. The ProliferationManager module is conceptualized as an overall container that, directly or indirectly, has the reference of everything involved within the proliferation model. It serves a bridge between the proliferation package and the supporting inventory/transport system. The InventoryHoldingPoint module at the bottom of the figure not only represents the InventoryHoldingPoint class (in the inventorylayer of the SupplyChain package), but also represents the whole inventory system which involve classes such as InventoryHoldingAbstract, Inventory, InventoryPolicyAbstract, Demand, DemandGenerator, DemandFillerAbstract, etc., as well as the transport system which may involve classes from the transportlayer such as DemandLoad, DemandLoadBuilder, TimeBasedLoadCarrier, etc. In the middle of the hierarchy, ItemCategoryAbstract and CItemType represent two important classes

that are specifically designed to model the proliferation ideas. The *CategoryLocationProliferator* enables the acceptance proliferation case and the *AdoptionProliferator* enables the adoption proliferation case.

In the simulation, the model will run on the inventory and transport layers, see Rossetti, Miman, and Varghese (2007), while the proliferators take effect by scheduling proliferation events on the model's event calendar. When the simulation clock reaches such an event, the proliferation procedure will take place, as follows: the proliferator (*CategoryLocationProliferator* or *AdoptionProliferator*) will notify the corresponding *ItemCategoryAbstract* about the proliferation occurrence. Then the *ItemCategoryAbstract* will execute the proliferation actions which includes updating the various reference sets and lists, generating the new *CitemType* object, etc., and passing the proliferation message to the *ProliferationManager*, which will in turn execute the physical modification on the inventory systems, such as setting up the inventory for the added item type in the related SSU(s) and the CSC and creating the *DemandGenerator* for the added item type, etc. For simplicity and space considerations, the full details of the Java implementation have been omitted, but are available upon request from the authors.

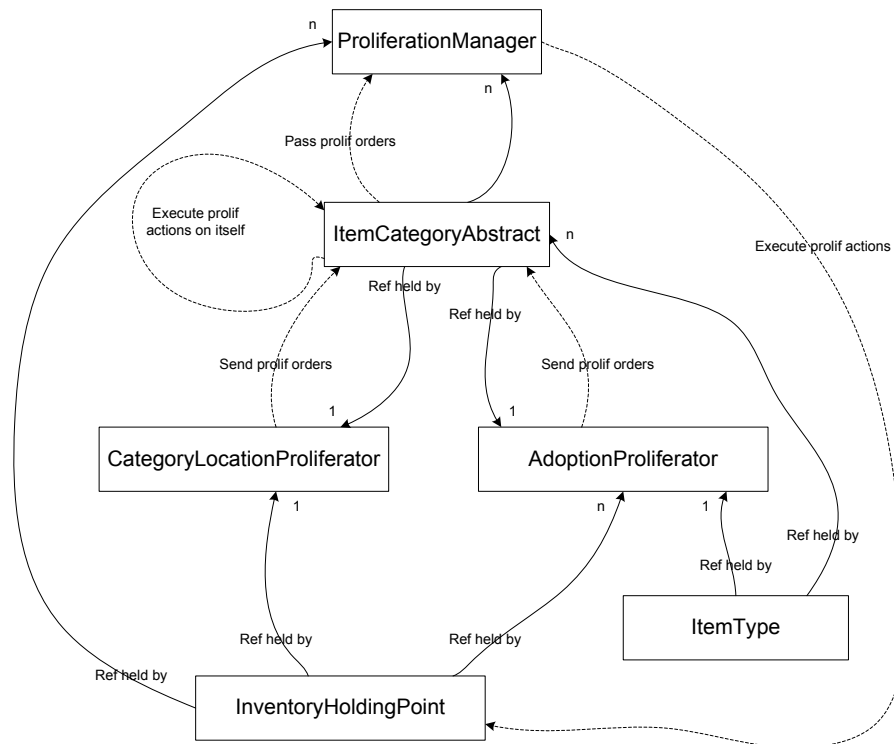


Figure 1: Proliferation Model Structure

4 EXPERIMENTATION AND RESULTS

The systems modeled in this paper are based loosely upon the Sisters of Mercy Health System. The Sisters of Mercy Health System is a hospital system based in St. Louis, Missouri that operates facilities and services in a seven-state area encompassing Arkansas, Kansas, Louisiana, Mississippi, Missouri, Oklahoma and Texas. The Health System is operated through regional hospitals called "Strategic Service Unit" (SSU). Presently Mercy operates ten major hospitals in the five adjoining states, providing healthcare services to millions of people who can easily access the state of the art medical technologies and harness the expertise of the best physicians in the area. In this section, we present a simple multi-echelon supply chain that was used for the basis of experiments to understand the effect of the different types of proliferation cases. In addition, the experiments and their results are discussed.

The network contains an external supplier, a CSC, 4 SSUs, namely, SSU_AR, SSU_MO, SSU_OK and SSU_KS, 4 LoadBuilders and a Carrier between the CSC and the SSUs. The base model contains 6 item Categories, namely, C1 ~ C6 with each Category having 1 item Type, namely T1. A category represents an independent demand stream. We assume that the demand follows a stationary Poisson process. The cardiac stent is an example of a Category but when we come to a specific stent, it must be of a certain item Type, such as a brand.

We classify the 6 Categories into 3 groups according to the demand rates for them, namely: Low, Medium and High, with each group containing 2 Categories. Within each group, we further assign different cubes to each category. All other attributes of the Category are the same (identically distributed). The settings for Category attributes are listed in Table 1. Note that for the Cube and Supplier Lead Time, the listed entries are the parameters (i.e. (minimum, most likely, maximum)) for Triangular distributions. For the Demand Rate, the listed entries represent how many units per day.

We adopt the By Quantity load forming policy. Specifically, we form the load by the cube with (min, max) policy: when the aggregate cube in the load queue reaches up to the minimum value, a load is formed by de-queuing the items until the queue is empty or until the total cube of the load reaches the maximum value, whichever comes first. Across the experiments, we adopt (500, 4000) cubic feet as the basic load forming limits. The transport cost rate is set to \$30/hour and the distance between each location is shown in Table 2. The cost segments are listed in Table 3. The last three items are annual rates.

Table 1: Item Category Attributes in the Pilot Experiments

	C1	C2	C3	C4	C5	C6
Cube	(5, 15, 25)	(30, 40, 50)	(5, 15, 25)	(30, 40, 50)	(5, 15, 25)	(30, 40, 50)
Demand Level	Low Demand		Medium Demand		High Demand	
Supplier Lead Time	(8, 10, 12)		(5, 7, 9)		(2, 4, 6)	
CSC Inventory (r,Q)	(8, 16)		(40, 80)		(120, 240)	
SSU Inventory (r,Q)	(2, 4)		(10, 20)		(30, 60)	
Demand Rate	0.1		1		10	

Table 2: Constant Transport Time (in hours)

From / To	CSC	SSU_AR	SSU_MO	SSU_OK	SSU_KS
Supplier	1	-	-	-	-
CSC	-	3.3	3.4	4.5	2

Table 3: Cost Settings

Cost Rates	Low Demand		Medium Demand		High Demand	
	C1	C2	C3	C4	C5	C6
Unit Price (\$)	100	300	10	30	1	3
Ordering Cost (\$)	10					
Capital Cost Rate (\$/\$)	10%		5%		1%	
Holding Cost Rate (\$/ft ³)	0.3		0.2		0.1	
Shortage Cost Rate (\$/\$)	200%					

Before proceeding to the proliferation experiments, let us clarify the common settings to be used in the experiments, so that we do not need to reiterate them for every experiment. In all the experiments, the performance metrics we primarily examine and compare include the Average SSU FR which is the simple average of the SSU_AR Fill Rate, SSU_MO Fill Rate, SSU_OK Fill Rate and SSU_KS Fill Rate, the Total Annual Cost, which is calculated based on the simulation output (e.g. average inventory levels, back order levels, number of transportations, etc) and the cost settings listed in Table 3, and the two segments of the Total Annual Cost, namely, the Annual Transportation Cost and the Annual Inventory Cost, where the Annual Inventory Cost is the sum of the annual warehousing cost, the capital cost and the shortage cost in the inventory system. In the system configuration and throughout the experimentation, the four SSUs are treated identically in terms of all aspects except for the transport time from the CSC, and from the preliminary experiments where the SSUs' individual fill rates are reported, we have already seen that there is negligible difference between them. Therefore, it is sensible to use the Average SSU Fill Rate to infer the SSU fill rate in general.

With the presence of SKU proliferation, the system will never achieve steady state, and all the performance metrics are time-horizon dependent. Therefore, it is critical that the simulation horizon (replication length) stays fixed for all the scenarios. Our research focus, combined with the result of preliminary experiments informs us that a 3-year horizon is a reasonable period to look at for a supply chain of this scale. So the replication length will be set to 3-years. In addition, a 50-day warm-up period is used for each experiment run. In the model, we allow the system to run without proliferation events during the warm up period so that the initial inventory level settings are allowed to be dispersed throughout the network.

In terms of the reported results, we present only the mean value (with 3 decimals for the fill rates) of the metrics and suppress the half-widths, which we control under 0.01 (for the fill rates') via manipulating the number of replications for the experiments. The experiment factors (control variables) are shown in bold in their respective experiment result tables.

4.1 Acceptance and Adoption

We are interested in seeing the relationships between the impact of the acceptance case and the impact of the adoption case to the supply chain and outlining any possible patterns. In this experiment, we fixed the compound proliferation rate ($\alpha + \beta$) to 0.005 and run 10 scenarios, each with a different combination of the acceptance rate (α) and the adoption rate (β). This bears realistic meanings. For example, the hospital management may allow a physician to request at most 1 new item type each year. With probability p , the physician may request an item type that is non-existent in the whole health system (the acceptance case will occur), and with probability $(1 - p)$, the physician may request an item type that has already been used in other hospitals within the health system (the adoption case will occur). In this case, p can be translated to $\alpha / (\alpha + \beta)$ in the SKU proliferation model where the $(\alpha + \beta)$ is fixed. The simulation results are presented in Table 4, as well as, Figure 2 and 3.

Table 4: Experiment Results for Different (α, β) Combinations

	α	β	CSC FR	Avg SSU FR	Annual Cost		
					Total	Transportation	Inventory
Scenario 1	0.005	0	0.880	0.917	32465	22386	10079
Scenario 2	0.0045	0.0005	0.892	0.904	31422	21559	9863
Scenario 3	0.004	0.001	0.901	0.903	31027	21239	9788
Scenario 4	0.0035	0.0015	0.910	0.905	31596	21960	9636
Scenario 5	0.003	0.002	0.920	0.909	31687	22182	9505
Scenario 6	0.0025	0.0025	0.928	0.911	30976	21702	9275
Scenario 7	0.002	0.003	0.936	0.917	31400	22749	8651
Scenario 8	0.0015	0.0035	0.946	0.931	32221	24290	7931
Scenario 9	0.001	0.004	0.957	0.939	33368	25859	7509
Scenario 10	0.0005	0.0045	0.968	0.942	33683	26762	6921

It is interesting to see that the CSC fill rate and the average SSU fill rate demonstrate very different response curves to the (α, β) combination. In particular, with the acceptance rate α decreasing (β increasing accordingly), the CSC fill rate keeps increasing which reveals roughly a reverse proportional relationship with α , while the average SSU fill rate first decreases and then increases, exhibiting a minimum point at scenario 3, i.e. $(\alpha, \beta) = (0.004, 0.001)$. The total cost also achieves its lowest point at scenario 3 and afterwards keeps increasing mildly -- the range of cost changes across the 10 scenarios is relatively small.

In order to examine what happens at scenario 3, we collected the average number of stockouts at SSU_AR (SSU_AR is chosen arbitrarily among the 4 SSUs because the four of them are almost identical) for each of the 6 item categories. The plot is shown in Figure 3. Note that the averages reported here are the across replication averages over the entire replication length (3 years) instead of the annual averages. It can be seen that the stockout is a rare event since the average number of occurrences is below 0.5 for all 6 categories. The plot exhibits surges at scenario 3, at least very obvious for category 1 and 6, which explains the dip points of the Average SSU Fill Rate and the Total Annual Cost.

Back to the physician's choice example in the beginning of this experiment, it seems that the least desirable p value is $0.004 / (0.004 + 0.001) = 0.8$ in the given system context.

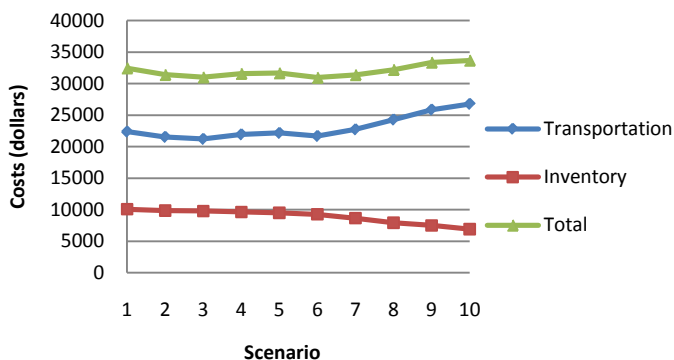


Figure 2: Cost vs. (α, β) Combinations

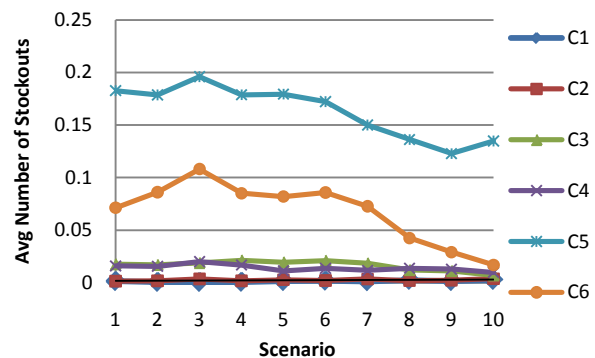


Figure 3: SSU_AR Stockouts vs. (α, β) Combinations

4.2 Response Surface Over (α , β)

It will be interesting to see the aggregate effect of the two factors (i.e. α and β) in a complete picture. For this purpose, we perform a series experiments consisting of 25 scenarios. These scenarios construct the complete enumeration of (α , β) pairs in the discrete range of [0.002, 0.01] with an increment size of 0.002. Five responses were selected, namely, the CSC Fill Rate, the Average SSU Fill Rate, the Total Costs, the Transportation Cost Segment and the Inventory Cost Segment. Again, all the costs are annual costs. The response surface for each of the five responses are plotted, as shown in Figure 4 to 7.

In Figure 4, the CSC Fill Rate is dominated by the acceptance rate α , and is relatively independent of the adoption rate β . The Average SSU Fill Rate is also dominated by α , while the effect of β is also visible in Figure 5. When α and β are both in effect, it is hard to discern which one is dominating the total cost, as shown in Figure 6. The general trend is that with the α and (or) β increasing in the experimenting range, the total cost decreases. However, the lowest point is at (α , β) = (0.008, 0.01) instead of at (0.01, 0.01). This is not surprising because we have known (from the previous experiments) that the two components of the total cost, i.e. the transportation cost and the inventory cost, display opposite trends with respect to the proliferation rates (both α and β). This is verified in Figure 7 and 8. In Figure 7 (the transportation cost surface), decreasing patterns are exhibited along both the α axis and the β axis, at all fixed levels of β and α , respectively. From the color of the areas on the surface, we can see that α factor has greater effect on the transportation cost than does the β factor. Similarly, monotone increasing patterns are exhibited in Figure 8 (the inventory cost surface) but the effect of β is obviously weaker than in the transportation cost.

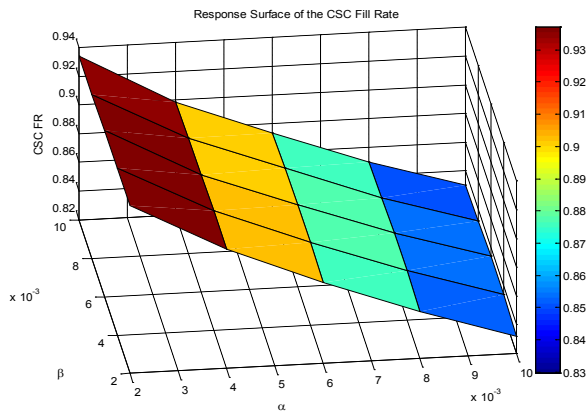


Figure 4: Response Surface of the CSC Fill Rate

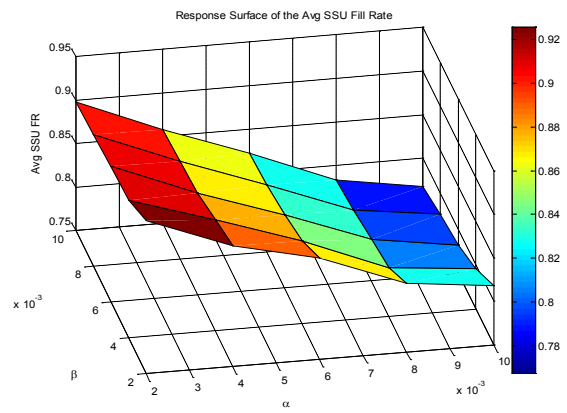


Figure 5: Response Surface of the Avg SSU Fill Rate

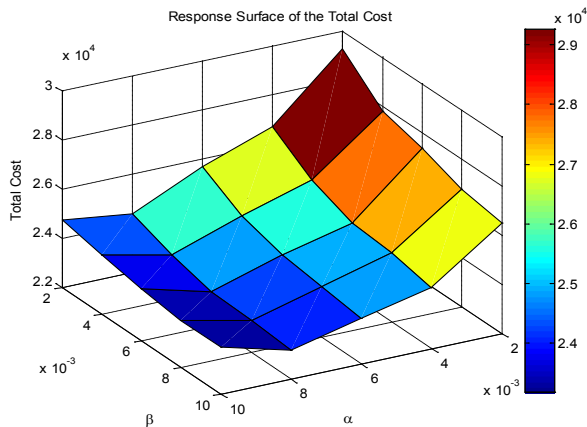


Figure 6: Response Surface of the Total Annual Cost

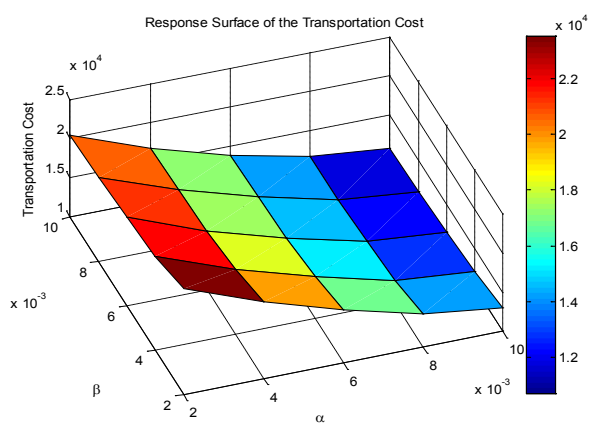


Figure 7: Response Surface of the Transportation Cost Segment

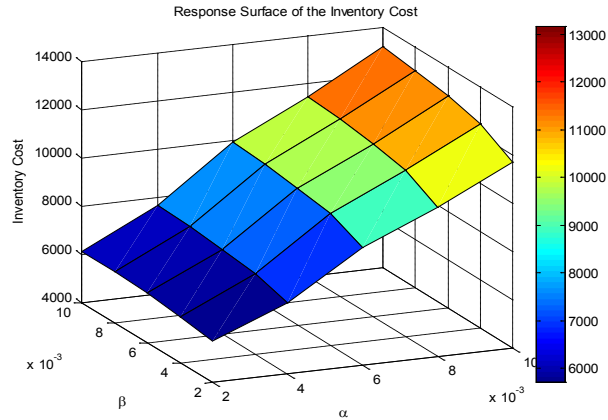


Figure 8: Response Surface of the Inventory Cost Segment

5 CONCLUSION AND FUTURE WORK

This paper investigated the SKU proliferation phenomenon in the health care supply chain. There are two types of proliferation paradigms in the multi-echelon supply environment, namely, the new SKU acceptance case and the existing SKU adoption case. The acceptance case indicates that when a hospital (or a particular physician in the hospital) places a request for a new item, this item is non-existent in the whole hospital system, while adoption means that the newly requested item does not exist in this particular hospital but does exist in some other hospitals within the same system. The acceptance and the adoption cases jointly affect the supply chain performances of the hospital system in terms of the service level (demand fill rate) and the cost. Primary findings include:

1. When the compound rate of proliferation ($\alpha + \beta$) is fixed at a reasonable level, the CSC fill rate and the average SSU fill rate demonstrate opposite response curves. In particular, with α decreasing (β increasing), the CSC fill rate exhibits a reverse proportional relationship with α , while the average SSU fill rate first decreases and then increases, having a minimum point somewhere in the middle, roughly at $(\alpha, \beta) = (0.004, 0.001)$. These observations indicate that when the physicians are given a fixed number of opportunities to request new item types, the ratio of the number of acceptance requests and the number of adoption requests will have an effect on the service levels. Specifically, the minimum point in the SSU fill rate curve should be avoided.
2. At a fixed acceptance level, increasing the adoption rate can slightly improve the CSC fill rate because the CSC can more easily supply the new product to more SSUs. However, increasing the adoption rate will adversely affect the SSU fill rates. In this sense, the CSC and individual SSUs have opposite objectives, therefore, careful negotiation and coordination work is necessary.

Many insights about the SKU proliferation were drawn from the investigation. However, we have made quite a few assumptions about the system characteristics and questions may exist regarding why those assumptions are proper and what can be done in the future. For example, in building the model we assumed that the demand rate for a category is fixed and all the item types within the category split the demand stream equally. We also assumed that the item types within the same category share the category's inventory space equally. These "equality" assumptions are useful in this paper because our purpose here is to investigate the SKU proliferation in the pure sense and the assumptions help to eliminate irrelevant randomness. However, in reality those assumptions may not hold. For example, it is easy to conjecture that newly accepted item types have higher popularity and thus have higher demand rates as well as higher adoption rates than their "older" counterparts in the category, so they must also enjoy more shelf space in the inventory in order to keep a satisfactory fill rate. In such cases, the "equality" assumptions in the demand rate do not hold and the investigation of how to adjust (optimize) the inventory sharing rule in accordance with the unequal demand splitting reality might be of interest in future work. Also of interest is obsolescence and its interaction with the proliferation. Obsolescence is the contrary of proliferation and is a natural resort to keep the supply chain stable. As technology advances, more and more item types will be invented and brought into the inventory (the SKU proliferation process). This has to be accompanied by the process of the obsolete item types being removed from the inventory. Otherwise, the size of the inventory will continue to increase. Therefore, incorporating the obsolescence process in the SKU proliferation phenomenon is a both interesting and necessary future work direction. Besides, if the obsolescence process is considered, the system stability may also be studied via more analytical approaches such as stochastic modeling, Markov chains, etc.

REFERENCES

- Abernathy, F. H., J. T. Dunlop, J. H. Hammond, and D. Weil. 2001. Control Your Inventory in a World of Lean Retailing. *Harvard Business Review*, November-December 2000.
- Andel, T. 2008. SKU proliferation and other knotty problems. *Modern Material Handling*, <http://www.mmh.com/blog/700000470/post/580022458.html>, accessed in September, 2008.
- Cook, M. 2001. The complexity of managing complexity. *Bain & Company*, http://www.bain.com/bainweb/LocalOffices/in_the_news_detail.asp?office_id=121&language=1&menu_id=127&id=3168&menu_url=office_publications.asp, accessed in June, 2007.
- Kelton, W. D., R. P. Sadowski, D. T. Sturrock. 2003. *Simulation with Arena*. McGraw-Hill Higher Education.
- Kravitz, R. 1998. Patient satisfaction with health care – critical outcome or trivial pursuit? *J Gen Intern Med.*, 13(4), April 1998, 280 – 282.
- Kukreja, A., C. P. Schmidt, and D. M. Miller. 2001. Stocking decisions for low-usage items in a multilocation inventory system. *Management Science*, Vol. 47, No. 10, October 2001, pp. 1371 – 1383.
- Lee, H. L. 1987. A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Management Science*, Vol. 33, No. 10, October 1987, pp. 1302 – 1316.
- Montgomery, D. C. and G. C. Runger. 2006. *Applied Statistics and Probability for Engineers*, 4th Edition. John Wiley & Sons, Inc.
- Montgomery, K. and E. S. Schneller. 2007. Hospitals' strategies for orchestrating selection of physician preference items. *The Milbank Quarterly*, Vol. 85, No. 2, 2007, pp. 307 – 335.
- National Center for Health Statistics. 2007. Inpatient Procedures. <http://www.cdc.gov/nchs/FASTATS/insurg.htm>, accessed in June 2007.
- Robinson, L. W. 1990. Optimal and approximate policies in multiperiod, multilocation inventory models with transshipments. *Operations Research*, Vol. 38, No. 2, March -- April 1990, pp. 278 – 295.
- Rossetti, M. D. 2008. JSL: An Open-Source Object-Oriented Framework for Discrete-Event Simulation in Java, *International Journal of Simulation and Process Modeling*, vol. 4, no. 1, pp. 69-87.
- Rossetti, M. D., B. Aylor, R. Jacoby, A. Prorock, and A. White. 2000. Simfone: An object-oriented simulation framework, *The Proceedings of the 2000 Winter Simulation Conference*, ed. J. Joines, R. Barton, P. Fishwick, and K. Kang, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc., pp. 1855-1864
- Rossetti, M. D., and V. Desai. 2007. Evaluating clustering methods for multi-echelon (r, Q) policy setting, *Industrial Engineering Research Conference Proceedings*, May 19-23, 2007, Nashville, TN.
- Rossetti, M. D., M. Miman, and V. Varghese. 2007. An object-oriented framework for simulating supply systems, *Journal of Simulation*, vol. 2, pp. 103-116.
- Rossetti, M.D., and Y. Tee. 2002. A robustness study of a multi-echelon inventory model via simulation, *International Journal of Production Economics*, No. 80, 265-277.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under award #0436687. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

AUTHOR BIOGRAPHIES

MANUEL D. ROSSETTI is an Associate Professor in the Industrial Engineering Department at the University of Arkansas. He received his Ph.D. in Industrial and Systems Engineering from The Ohio State University. He serves as an Associate Editor for the International Journal of Modeling and Simulation and is active in IIE, INFORMS, and ASEE. He was a WSC proceedings editor in 2004 and a co-editor for the WSC 2009 conference. He is also author of the textbook, *Simulation Modeling and Arena* published by John Wiley & Sons. His email is <rossetti@uark.edu>

YANCHAO LIU is currently a Ph.D. student in the Department of Industrial and Systems Engineering at the University of Wisconsin-Madison. He received his M.S. in Industrial Engineering from the University of Arkansas and B.S. in Industrial Engineering from Huazhong University of Science and Technology, China. His research interests include simulation output analysis, transient analysis of Markov processes, and life cycle modeling of complex systems.