An object-oriented supply chain simulation for products with high service level requirements in the embedded devices industry

Detlef Herbert Grittner*

Sohard Software GmbH,
Wuerzburger Str. 197,
90766 Fuerth, Germany
E-mail: detlef.grittner@t-online.de
*Corresponding author

Raul Valverde

John Molson School of Business,
Concordia University,
1455 de Maisonneuve Blvd. W.,
S-MB 11311, Montreal, Canada
E-mail: rvalverde@jmsb.concordia.ca

Abstract: The supply chain of a case study in the embedded devices industry was examined from the perspective of a high service level for the delivery of one of its products. For this purpose, an agent-based object-oriented model of the supply chain with a quantity reorder system for the inventory management was developed. The historical demand data from the ERP system of the case study was examined and pseudo-random numbers for a Monte Carlo simulation of the supply chain was generated with it. The paper examines the performance of the supply chain by using a simulation with a stockout penalty and the percentage of items delivered from stock. Results from simulation show that no single solution for the reorder point and quantity with an optimal stockout penalty can be found. The simulation generated a solution for the minimum average penalty.

Biographical notes: Detlef Herbert Grittner is a Senior Software Developer with Sohard Software GmbH in Germany where he develops software applications for the healthcare industry. He worked many years in the development and architecture of software for medical devices, middleware for distributed systems and image processing systems. He holds a Master of Science in Information Systems Management from the University of Liverpool in England and specialising in the development of software applications for supply chain systems. His research interests include software architecture, development with Java, JEE, C++, Linux server systems and open source systems.

Raul Valverde is Lecturer in Management Information Systems and Supply Chain Operations Management and Academic Director of the Business Technology Program at the John Molson School of Business of Concordia University in Montreal Canada. He holds a Bachelor of Science in Mathematics and Management from Excelsior College (US), Master of Engineering in Electrical and Computer Engineering from Concordia University (Canada), Post Master of Business Administration from McGill University (Canada) and Doctor of Business Administration in Information Systems from the University of Southern Queensland (Australia). He is also a Registered Professional Engineer in Canada and a Public Accountant in Australia. He has more than 17 years of professional experience in information technology, mathematical modelling, financial analysis and programming. His main research interests include supply chain systems, risk management, e-business, information security and auditing, accounting and financial information systems, fraud detection and reengineering.

1   Introduction

The case study for this research is the inventory management system of a specialised network card component of a company that produces embedded devices for specialised applications. The company outsources the production of this component to several suppliers. The response time of the supplier is critical and a high level of service is required.

There are few assumptions that are made for this study: the transportation cost is neglected, because the network cards can be bundled into deliveries with other products, the inventory holding cost is assumed to be rather small and only the opportunity cost of about 7% interest rate for the capital bound in stock plays a role in the calculation.

Up to now the approach to this problem has been to keep concise records of the historical demand and delivery times. Based on these it was possible to estimate the needed inventory level and the performance of the supply chain. This case study examines how far this approach was successful and researches another way of estimation and prediction by using simulation of the supply chain.

First, it was necessary to formalise the existing heuristic used for the decision making of the managers and, second, a model had to be established that reflects the existing supply chain and its decision making parameters. This resulted in an agent-based model for a quantity-reorder system taking into regard the lead time for the supply. An object-oriented approach proved to be very successful in analysing the existing supply chain and setting up a model that could directly be implemented as a Java programme. For the realisation of the simulation this case study uses a very simple form of agent-based simulation with only two agents – the supplier and the retailer – and the demand is triggered by events from a generated demand dataset.

For the purpose of investigating possible scenarios of future demand, the historical demand data was analysed and similar, pseudo-random demand datasets generated that served as input for the model of the supply chain. The service level requirements were set in two ways: First, a penalty for stock outs was defined, with the goal to find a solution for the minimum stockout penalty cost. Second, the service level as the percentage of items delivered from stock was used and the results from the first approach were evaluated under this perspective.

2   Background and literature review

The term supply chain management has a broad meaning in economics. Chen and Paulraj (2004) observe, it is the relationship between members in the supply chain, that gets more important, "since suppliers have a profound and direct impact on cost, quality, time and responsiveness of the buying firms, the management of business and relationships with other members of the supply chain (i.e., buyer-suppler relationship) is increasingly being referred to as SCM". This contrasts SCM from mere inventory management that mainly focuses on an internal operation of the buyer or retailer. It is a very important observation of Chen and Paulraj (2004) "that focusing on a single element in the chain cannot assure the effectiveness of the whole system" and they come to a rather pessimistic evaluation of mathematical approaches to supply chain management, because these give a good "insight in well-defined supply chain settings involving few decision variables and highly restrictive assumptions". The assumptions are often unrealistic and not suitable for the complex situation of real supply chains (Chen and Paulraj, 2004).

Mathematical models for inventory management have their limitations. For example, Viswanadham et al. (2000) choose a stochastic approach with Petri nets and compare a make-to-stock system, that holds the end products in inventory, to an assemble-to-stock system, that holds parts of the end product in inventory and these are assembled when an order arrives. They find that for two end products called D and E in their study, a make-to-stock system "offers better serviceability in terms of faster access to end products D and E, therefore reducing the probability of back ordering them" (Viswanadham et al., 2000). Even though intuition might already suggest this result and the recommendation that make-to-stock systems are preferable "when delay related costs are substantial when compared to inventory costs" (Viswanadham et al., 2000), it is of importance that this observation holds in a mathematical model as well, because this puts confidence in the choice of such a system.

When choosing a model for inventory management, there are principally two systems. One is based on the quantity when to reorder and the other is a periodic system. Buxey (2006) examines under that conditions one of these models should be used. The so called 'Q-System' concentrates on finding a quantity of items on inventory and when the stock level falls to this value, then a reorder is triggered. The 'P-System' defines a period after which items will be reordered, regardless of the actual stock level. Furthermore Buxey explains that for "goods with erratic demand patterns or ordered infrequently" a Q-System is typically preferred (Buxey, 2006). A very important point is that for goods that have long time between orders and the logistics for scheduling the transport is not an issue, i.e., even small quantities can be send as parcel, then a Q-System is a suitable choice (Buxey, 2006). This influenced the choice of a quantity-reorder system for the case study.

Another important factor to consider for inventory management is the problem, where the store should be located. Wallin et al. (2006) describe possible approaches for the inventory management of purchased items including speculation that is the traditional form with the retailer holding the inventory, postponement, where the order is postponed until demand occurs. Consignments are created when the retailer holds the inventory, but the supplier remains the owner, or reverse consignment, where the customer holds the inventory, but the retailer remains the owner of the goods.

An alternative suitable for dynamic demand is not only to look at the historical demand but to adapt the forecast with current observations. Spedding and Chan (2000) propose a Bayesian model that can deal with conditional probabilities and therefore is adaptable to changes. At least the findings of their paper suggest that this alternative method might be suitable for modelling "the inventory demand for the high technology batch production environment" (Spedding and Chan, 2000). But one problem is which parameters should be chosen in order to influence the filter, because "the expert knowledge" (Spedding and Chan, 2000) that the authors want to use as input might not be available or easily quantifiable, so that it can be used as numeric value for the filter.

Kobbacy and Liang (1999) make the interesting proposal that the choice of the right inventory management system should be done by an expert system that uses statistical properties of the demand data and lead times, in order to find an appropriate system. Yet this approach needs models for the inventory management, from which the expert system can make a choice. And the approach is more or less static, that is a change of the analysed system, in order to improve the performance, is not part of the expert system.

In his research, Verma (2006) observes that simulations can validate mathematical models and demonstrates this with a three-tier supply chain and demand following a Poission distribution. The result shows that base-stock models are quite effective (Verma, 2006). One drawback seems to be that no real data was used for the simulation.

Another approach is mentioned in Tarim and Smith (2008) describing an (R, S) policy where "a replenishment is placed every R periods to raise the inventory position to the order-up-to-level S". This can help to remove nervousness from the system, i.e., the uncertainty of planning is reduced. This happens because the reorders come after regular periods. But it seems that it needs a rather good prediction of the demand during a period, otherwise the reorder quantities will vary largely and thus reintroduce uncertainty into the system.

De Sensi et al. (2007) examine a three echelon system that has producers, distribution centres and retailers using a simulation for validation. They use a review period-based inventory policy that seems appropriate for the beverage industry chosen for their study, and they search for the optimal review period in such a system. This case study also highlights the importance of looking at more than one aspect of a supply chain taking into regards inventory policies, lead times and demand intensity and variability as well (De Sensi et al., 2007).

Given a choice between many suppliers in the supply chain, Wadhwa et al. (2008) use simulations in order to find an optimal selection vendor to select. To achieve this it is necessary to have very fast information exchange, when there are dynamic changes in the supply chain parameters that the authors consider as vital for the success of a dynamic supply chain and they propose worldwide IT systems as solution (Wadhwa et al., 2008). Obviously this does not solve the problem that not all supply chain partners might be willing to be integrated in such a system that could even lead to their removal as a potential supplier under certain circumstances.

Cigolinia et al. (2011) worked on an object-oriented simulation model of supply chains. The proposed meta-model is made up from a user interface (to define the characteristics of the supply chain), an ad hoc objects library and a software application, to build the simulation model. With their research, Cigolinia et al. (2011) showed the benefits of an object-oriented simulation including the flexibility of configuration to test a specific supply chain within minutes given ability to reuse objects from an ad hoc objects library.

van der Zee and van der Vorst (2005) take the simulation approach even further suggesting that there should be a generic, object-oriented framework for modelling supply chain simulations. The goal would be to make simulations as a decision tool accessible to the stake holders and not only the analyst (van der Zee and van der Vorst, 2005). A very important aspect is not to choose a discrete event simulation as it is done in many other supply chain simulations (van der Zee and van der Vorst, 2005), but to rely on a agent-based system that promises better support for 'decision-making capabilities' (van der Zee and van der Vorst, 2005). As the authors themselves mention their work is limited in the sense that the proposed framework is not yet linked to simulation modelling methodologies (van der Zee and van der Vorst, 2005), but this would be necessary in order to get meaningful results with the simulation. Even though the simulation in this case study does not implement the suggested model in detail, it is strongly influenced by the idea to implement the participants of the supply chain as agents.

The Java language has been used successfully in the implementation of supply chain simulations. Anigbogu et al. (2011) developed a supply chain simulation with Java in their work 'An intelligent model for sales and inventory management'. The system developed had the ability of providing automatic demand and lead time pattern identification for inventory management based on fuzzy logic. The simulation was developed with Java 2 Enterprise Edition (J2EE).

Software implementations of inventory management systems simulations have been also implemented based on web-based programming languages. Karim et al. (2011) implemented a web-based system of managing inventories by using the PHP language and MySQL as a backend database with CSS implemented for the interface.

Even though the following description of the general Monte Carlo simulation, Landau (2000) originates in the field of physics, it gives a very understandable description of this method in general: The idea is to generate random numbers in order to model the stochastic behaviour of an input parameter, and each different sequence of random numbers causes another result. Stefanovic et al. (2008) describe that a Monte Carlo simulation is suitable to generate the demand as an external event to a system modelling a supply network. In this case study, the term Monte Carlo simulation is used to describe that many random demand datasets are used as input for a deterministic supply chain model that produces many results.

3   Supply chain model

A very common method for controlling inventory levels is the quantity-reorder model, that is a perpetual model, but is not based on a periodic review. Instead it uses the actual quantity of the inventory on stock and defines a so called reorder point. When this point is reached, then a reorder to the supplier is triggered.

That is, there are two important parameters, that are the reorder point and the reorder quantity. Both of these must be known in advance and a very common way to calculate these is based on the economic order quantity (EOQ). The EOQ model assumes that the stock is delivered instantly, when an reorder is triggered, and the reorder point is not taken into account for the simple model. When the inventory is depleted, then a reorder is triggered. Another important assumption is that production is infinite. That means an arbitrarily large order can be ordered and will appear immediately on stock.

The calculation uses the following parameters and formula:

D   demand per year

RC   reorder cost

HC   holding cost.

EOQ:

$$Q = \sqrt{\frac{2 \cdot RC \cdot D}{HC}}$$ (1)

The assumption that the production is infinite can be removed and there is an alternative formula for EOQ with finite production.

P   finite production per year

$$Q = \sqrt{\frac{2 \cdot RC \cdot D}{HC}} \cdot \sqrt{\frac{P}{P - D}}$$ (2)

It is important to note that this still assumes that the delivery will arrive immediately on stock, but in reality there will be a deviation. This is caused by setup times for machines, before the production can begin, the production time itself and transportation time until the items reach the retailer's inventory. The sum of these times is called lead time and needs to be accounted for.

A very simple model is to hold a safety stock that holds enough items to deal with the demand that occurs during the lead time. The following formula helps to calculate the reorder point when the lead time is constant and the demand is variable under the assumption that it is normally distributed:

ROP reorder point

t_l constant lead time

∂ mean demand per time unit

Z   number of standard deviation

σ standard deviation of demand

$$ROP = tl \cdot \partial + \sigma \cdot \sqrt{t_l} \text{ (Waters, 2003)}$$ (3)

For this simulation a supply chain model was developed with specific requirements. The supply chain model had a single supplier and a single product with one inventory. The algorithm for restocking was based on a quantity reorder system, i.e., a reorder point was defined and every time the stock level dropped below this value, the system issued a reorder of a specified quantity to the supplier. Contrary to the real world situation the simulation would choose a fixed reorder quantity and level. The heuristic that during a stockout a larger quantity than the mere reorder quantity should be added was slightly adapted. The principle underpinning the heuristic was to choose a reorder quantity that avoids an immediate stockout, when the resupply arrived and the backorder was fulfilled. A viable approach to simplify this was to add the demand of the backorder to the reorder quantity. Including stockouts and backorders the complete algorithm was as follows:

I inventory level

d quantity of items for demand

r reorder quantity

l reorder point (low quantity)

s items already ordered but not yet delivered (supply line)

o order to supplier.

The following formula indicated a stockout:

$$l - d < 0 \tag{4}$$

This was the check, whether a reorder should be issued:

$$l + s - d \leq l \tag{5}$$

Calculating the reorder quantity used the following formula:

$$o = r + d \tag{6}$$

No partial deliveries to customers were allowed; instead the retailer delivered the complete order on a later date. The heuristic of adding the demand of the backorder to the reorder quantity was formalised as follows: The idea behind the formula for the reorder was that the inventory would not be used up completely, when the delivery from the supplier arrived and the backorder fulfilled immediately after that. So the safety stock would be reached despite the unanticipated demand that caused the backorder. Furthermore, no lost demand should occur, but the system should fulfil all demand.

This algorithm had one side effect, when the reorder quantity is lower than the reorder size, because then several reorders would be triggered, because s would not be large enough after the first reorder, so that (5) would still hold true. Nevertheless the algorithm was tested as is, because the result would show, if the intuition, that triggering several reorders with one reorder point was disadvantageous, was true.

The simulation was limited to 104 calendar weeks and it could happen that demand would only be fulfilled outside the simulation time. In this case the demand was recorded as lost nevertheless, but solutions with lost demand were rejected during the analysis.

The production lead times consisted of two elements, first a setup time that was needed before the production could begin and the production time for each item.

t_s setup time for production

t_l lead time per item

q number of items ordered

T the time until the delivery arrives at the supplier.

Then the total time for the production and delivery was calculated as:

$$T = t_s + t_l \cdot q \tag{7}$$

Based on the aforementioned heuristic approach, the simulation offered another option that is splitting the deliveries from the supplier in several smaller, partial deliveries. Very intuitively, this would reduce the demand during lead time, but the setup cost for the order would apply only once. Disregarding the additional transportation cost, that was neglected in this case study, a part of the items would arrive in a much shorter time at the retailer, of course for completion of all partial deliveries still the original total time applied.

When the overall time for the delivery of the complete quantity of items per order remained the same and the order itself is split into several, partial deliveries, then the total reorder quantity q could be described in a continuous way as:

$$\int q_t = q \text{ for each time interval } t \text{ a part of the order is delivered.}$$

This had to be applied to the discrete order quantities of the given product and it was assumed that the orders are split by dividing by the 'order-split-factor' n:

$$\left(\sum_1^{n-1} \frac{q}{n} \right) + \left(\frac{q}{n} + q \bmod n \right) = q \tag{8}$$

or

$$(n-1)\frac{q}{n} + \left(\frac{q}{n} + q \bmod n \right) = q \tag{9}$$

$q, n \in \mathbb{N}$ and mod is the modulo operator for divisions.

The formula explicitly displays the last order, that includes the remainder of the integer division, that results from the constraint, that if $q, m, n \in \mathbb{N}$, then

$$m = \frac{q}{n} \wedge (q \bmod n) > 0 \Rightarrow m \cdot n \neq q.$$

This remainder was not set up as a separate delivery, but added to the last delivery, that is a simplification of the algorithm, and the idea is to avoid deliveries where the delivered quantity at time i was lower than a partial delivery so the following constraint applied:

$$q_i \geq \frac{q}{n}.$$

The overall delivery time stayed the same as with not split orders. The lead time for the last order with the added remainder is:

$$t_r = t_i \cdot \left(\frac{q}{n} + q \bmod n \right) \tag{10}$$

The lead time for each of the other $n - 1$ partial orders is:

$$t_p = t_i \frac{q}{n} \tag{11}$$

With (10) and (11) it follows that the complete lead time for all partial deliveries is:

$$t_s + \left(\sum_1^{n-1} t_p \right) + t_r = t$$
$$t_s + (n-1)t_p + t_r = t \tag{12}$$

The setup time per order is t_s and the lead time per single item is t_i. Therefore, after the period t_s every time period t_p a partial delivery of at least $\dfrac{q}{n}$ items arrives, with the exception of the last delivery with $\dfrac{q}{n} + q \bmod n$ items.

With setting (10) and (11) in (12) it follows:

$$t_s + (n-1)t_i \frac{q}{n} + t_i \cdot \left(\frac{q}{n} + q \bmod n \right) = t$$

$$t_s + t_l \cdot \left((n-1)\frac{q}{n} + \left(\frac{q}{n} + q \bmod n \right) \right) = t \text{ and according to (9)}$$

$$t_s + t_l \cdot q = t = T \tag{13}$$

This reflected the math behind the algorithm used in the simulation in order to calculate the time for each partial delivery and proved that the total time stayed the same.

The simulation implemented this scenario with an object-oriented approach, where the supplier and the retailer act as agents, each realising a set of logical decisions and triggering the flow of the goods from one agent to another.

The retailer agent could:

- fulfil an order from inventory

- get the reorder quantity from an inventory policy

- if it had not enough stock, then it reordered supply from the supplier and put the order into a backorder list

- tell the supplier to split orders based on a given factor

- upon resupply increase the stock of the inventory

- on each step in the simulation check whether the backorder could be fulfilled from the inventory at hand.

The supplier agent was able to:

- receive orders

- split them with a given factor into n orders according to (8)

- put each order in a production list

- check on each step in the simulation if one of the items in the production list was due for delivery and send it to the retailer.

The retailer itself did not know directly about the inventory policy, but accessed it through an interface that would allow for replacing the policy itself in an expanded version of the simulation. The simulation only realised a quantity-reorder policy though

that checked for a stockout as in $l + s - \mathrm{d} \leq 1$ (5) and if this held true then the reorder quantity was calculated with $o = r + d$ (6).

The following UML class diagram shows the classes of the aforementioned agents and how they were related to each other:

Figure 1   UML class diagram of supply chain simulation
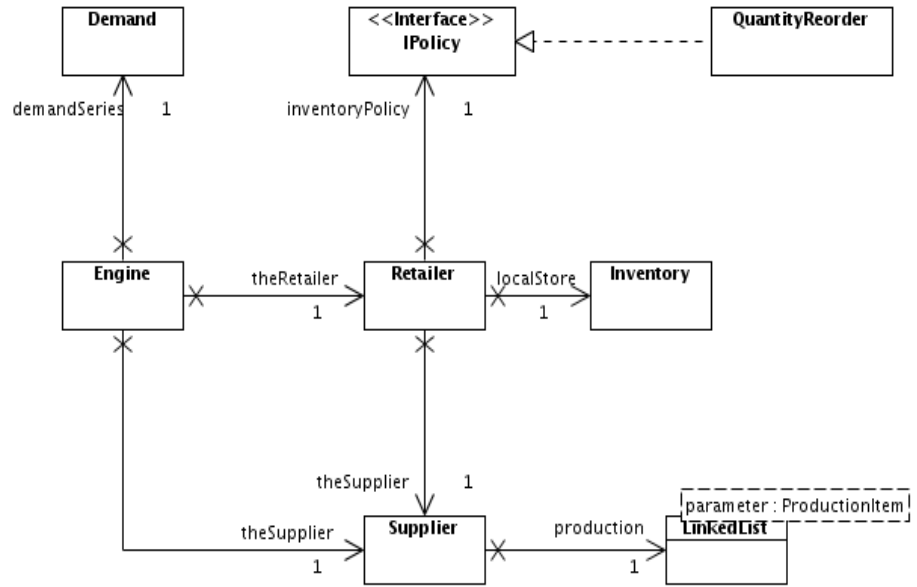
The retailer object was responsible for keeping the Inventory object *localStore* up to date. For calculating the reorder points the class implementing the *IPolicy* interface was responsible and in the actual solution there was only one implementation that was a quantity-reorder model as reflected in the class name.

The engine was responsible for running the timely correct behaviour of the simulation. It checked the object *demandSeries* of class demand. The class demand contained all customer orders and their specific point of time in the simulation. When there was a demand, then the instance of engine would send it to the retailer instance. This retailer checked the *localStore* and if not enough items were present, calculated a order quantity for the supplier by using the inventory policy accessible through *IPolicy*.

Another important aspect of the model is that the supplier supported a production list. Every time an order arrived, the point of time in the simulation, when it was ready for delivery to the retailer was calculated. The engine would trigger the supplier at every step in the simulation and it checked the production list for due deliveries to the retailer and these were then issued directly to the retailer.

Since it was possible to split orders into several deliveries from the supplier, each order was split by the order-splitting algorithm with the splitting factor n provided by the retailer. This algorithm was implemented by the *Supplier* class. The implementation calculated for each partial delivery the time, when it was due for delivery, and put the result as a separate order into the production list. That is, for split orders instead of 1 altogether *n* orders stood in the list.

This resulted in an agent-based simulation model, because each agent could hold its own internal state, e.g., the retailer had a backorder list and the supplier a production item list. The time unit was defined one minute as defined in Table 3, but state changes could only occur when at the specific point in time an order was found in the demand instance, a back order in the retailer instance or a pending order in the list of production items. The reason for choosing this approach was that each class, specifically *retailer* and *supplier* could easily be unit tested. At the same time, this approach allows for implementing complex behaviour in each agent without breaking the encapsulation of the classes in the model.

The time the customer had to wait for the delivery, could be measured in minutes during the experiment. Then a hypothetical cost for the delayed items was the number of items multiplied by the time needed for the delivery. This calculation served as a measurement for the penalty the business would suffer, when it did not meet the demand in time.

The definition for the penalty calculation is:

p penalty for the stockout

q number of items delayed

Δt delay of delivery in minutes

$$p = q \cdot \Delta t \tag{14}$$

4 Simulation design

The following parameters were made configurable in the simulation:

- lead time per item
- setup time per order
- splitting factor of the orders to the supplier.

A matrix for the chosen parameters and experiments was set up:

Table 1   Parameters chosen for experiments

Experiment	Setup time per order	Lead time per item	Splitting factor
N	2,400	56	1
S	2,400	56	4
NcL	2,400	28	1
ScL	2,400	28	4
ScLS	1,200	28	4

where the abbreviations of the experiments stood for:

N   normal

S   split supplier order

NcL   normal with compressed lead time

ScL   split supplier order with compressed lead time

ScLS   split supplier order with compressed lead and setup time.

The simulation implemented a Monte Carlo simulation with a pseudo-random set of demand data as input for an iterative algorithm with certain parameters for the inventory management. The demand data was randomly generated with the statistical properties of the original demand data. The experiment consisted of 100 randomly generated datasets and for each the iterative algorithm gave a result set.

The model was a quantity-reorder model with a reorder point and a reorder quantity. The reorder quantity and the reorder point were discrete values, therefore it was possible to iteratively calculate all solutions within the following constraints:

$2 <=$ reorderquantity $r <= 500$

$2 <=$ reorder point $l <= 500$

These constraints reflected limits of the storage place and the average, historical demand per year as an upper boundary. When for both parameters the value 500 was chosen, then the inventory theoretically could reach a limit of 1,000 items without additional backorders. This was already more than the average demand per year and it was assumed that such an overstocking was outside of a feasible solution.

The values for the lead time per item were chosen based on the observation of the historical data that suggested lead times between 35 min and 65 min per item. The first value of 56 minutes represented a weighted average of these values with an emphasis on the higher one. Since there seemed to be an improvement in the historical lead times, this was reflected in another simulation run with a lead time of 28 min, that was the half of the first value.

Since the setup time was another changeable parameter, there was one experiment with a value reduced from 2,400 min to 1,200 min. Actually, this parameter set was chosen based on the results from the other experiments.

Figure 2   Monte Carlo simulation of supply chain
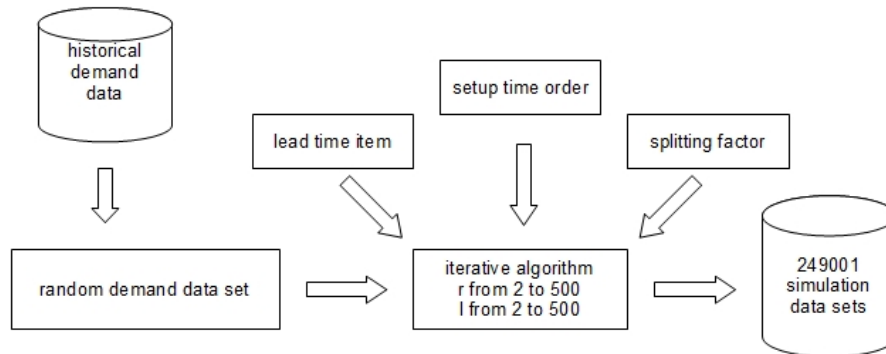
An important aspect of each Monte Carlo simulation model is the sampling of the randomised input data. In this case the demand data, because it was uncertain, had to be randomised.

To obtain randomised sets of historical demand data an algorithm implemented by the Apache Jakarta project was used. It is realised by the class EmpiricalDistributionImpl of the org.apache.commons.math package, that implements "a variable kernel method with Gaussian smoothing" (Apache.org, 2008). This is based on the work of Silverman (1986) and the advantage of the method is that it takes into account the neighbourhood of each point in the sample. This verifies that the estimator will preserve properties of the order of the values.

5 Research design

The aim of the research was to find out whether the results from the simulation could be used for estimating an appropriate solution for the reorder point and quantity. Since all solutions within given constraints were calculated, it should be possible to find an optimal solution within these. This step is often referred to as the aggregation of the results of a stochastic simulation based on a Monte Carlo simulation.

The approach used in the research was to find an optimum across all solution of all demand samples; this is called the cross-sample analysis here. In this case the minimum average service level penalty over all samples was examined. Then it was evaluated how many outliers there were from the mean plus a 95% confidence interval using $+- 2\sigma$. The values below that were disregarded, because these would mean a service level penalty cost below the mean and this was even beneficial, but the values above the confidence interval were problematic. Furthermore the percentage of the results that fulfil a 95% service level were also taken as metric.

5.1 Analysis method cross-sample

With this approach, the service level over all 100 tries for the demand samples for each reorder point and quantity was taken. Then it was possible to search for the minimum mean unit-minute value over all 249,001 solutions. The reorder point and quantity that led to the minimum mean value of the service level penalty, could be taken as a solution that on average incurred the least cost. The service level penalty calculated as unit-time was minimised and when there were several equal solutions that with the minimum inventory unit-time value was chosen. This approach run across all simulations of the demand samples as shown in Figure 3:

l   the low quantity, i.e., the reorder point

r   reorder quantity.

There were 100 values for the dataset with the mean minimum value for the stockout costs. Since the minimum mean was the cost to be expected on average, it was preferable to have lower, but not higher values. Therefore for assessing the solution the upper limit of a 95% confidence interval based on the 0.975 quantile of the normal distribution was taken.

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$\overline{x}$ the arithmetic mean

x_i the unit-time value for the experiment of the ith demand data sample for the given quantity and reorder point

N the number of experiments, 100 in this case study

$$s = \sqrt{\frac{1}{(N-1)}\sum_{l=1}^{N}(x_i - \overline{x})^2}$$

s the standard deviation from the sample of unit-time values

$$SE_{\overline{x}} = \frac{s}{\sqrt{N}}$$

$SE_{\overline{x}}$ standard error of the mean.

Figure 3 Cross-sample method
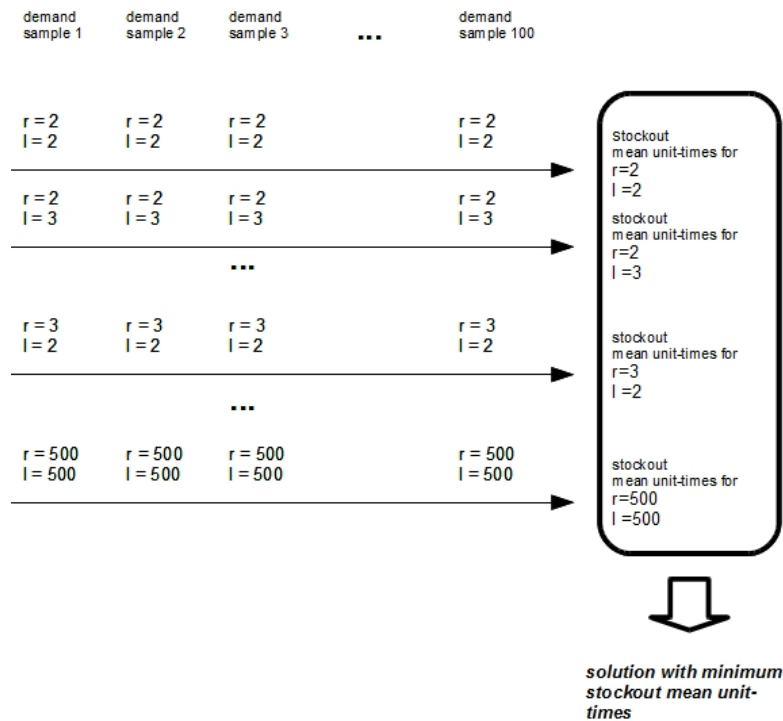
Upper bound of the 95% confidence interval:

$$Cu = \overline{x} + 1.96 \cdot SE_{\overline{x}}$$

Lower bound of the 95% confidence interval:

$$Cl = \overline{x} - 1.96 \cdot SE_{\overline{x}}$$

The confidence interval typically marks the boundaries into that the values of a sample should fall, so that the mean value can be regarded as an appropriate estimate for all samples. Under the assumption that the values are normally distributed and that 95% of the values should lie within the mean value and the limits marked by the standard error, the above formula for the upper and lower limit of the confidence interval apply.

The values above the prediction with the mean and the expected error marked by the upper bound of the confidence interval were a problem.

The values below and up to *Cu* should be more frequent than those above *Cu* for a good solution, i.e., *Cu* would give a good estimate for the maximum cost expected from the stockouts in the solution for reorder quantity and point. Therefore the analysis showed the frequency in bins from 0 up to *Cu* and *Cu* to the maximum value.

Furthermore, the confidence interval can be used for comparing different samples from a population. In this case the results for the minimum mean stockout penalty for a pair of reorder point and quantity was taken and tested against other demand data. The mean values of all samples should lie within the confidence interval calculated from the other samples. The study uses this in order to show that a special parameter set for reorder point and quantity has a mean value for the penalty cost that lies within the confidence interval even for different demand datasets. This is serves as a validation of the found solution.

6 Historical data collection

The historical demand data showed long intervals of inactivity and typically there were only a few orders per calendar week. Therefore the data was sampled in calendar weeks, i.e., the activities during the working days of one calendar week were sampled into one demand sample.

Actually, this did not weaken the conditions, but quite contrary tightened them, because in the chosen model every order of the demand had to be delivered in one complete lot. When there were three orders in a calendar week, then each smaller order could be delivered in a shorter time. But with one larger order summed up from the smaller ones it was more likely that a stockout could occur. Nevertheless, this was not a problem, because a system that can handle such a load would be more robust.

This kind of sampling only applied to the simulation data and the calculation of the reorder quantities and reorder points. The evaluation of the service times was done with the separate orders. For example if there were two orders with number 1 at 2006-06-27 and number 2 at 2006-06-29, then both occur in the calendar week 26 of the year 2006. For evaluating the service times of the historical data two orders were assumed, both occur in calendar week 26, and if number 1 could be delivered from stock and number 2 only one calendar week later, then the total service time in unit-times would be:

$$0 \cdot \text{quantitynumber } 1 + 1 \cdot \text{quantitynumber } 2.$$

Comparing this to the approach with sampling the orders into one demand sample per calendar week, it would be different, because now there was only one demand, that is the sum of quantity number 1 and quantity number 2. Since number 2 already caused a stockout, any larger order would do as well, and if the delivery for the combined order took 2 weeks then the service time in unit-time was:

$2 \cdot (\text{quantitynumber } 1 + \text{quantitynumber } 2)$.

As it later showed, this did not put the real supply chain at any advantage during the analysis, even though the criteria for the simulation were tighter. And of course the historical data was sampled on a daily basis for orders and calendar weeks for the delivery. So any estimates and comparisons based on the aforementioned system seemed accurate enough. The data sampled on the basis of calendar weeks showed the following properties that were used for the calculation with the analytical formula:

These were the absolute demand data in items per year and statistical properties for the demand data sampled in calendar weeks:

The statistical values over all four years were:

Table 2   Demand data per year

Year	Absolute demand	Average demand per year	Mean (per calendar week)	Standard deviation
1–4	2,613	653	12.56	23.44

Table 3   Demand data over four years

Year	Absolute demand	Mean (per calendar week)	Standard deviation
1	621	11.94	15.11
2	666	12.81	21.42
3	596	11.69	22.12
4	730	13.77	32.26

Concerning the holding cost, the rent for the storage space or the cost of electricity, etc., could be neglected compared to the opportunity cost of the capital bound in the inventory. It was assumed that the capital could be invested with a market interest rate of 7% per year and that this earning would be lost when the capital was invested in inventory instead. This applied as long as the products were not sold, therefore the holding cost per year was set as 7% of the reordering cost. With the EOQ formula (1) it follows:

$$D = 653$$
$$RC = 50$$
$$HC = 0.07 * 50 = 3.5$$
$$Q = \sqrt{\frac{2 \cdot 50 \cdot 653}{3.5}} \approx 137$$

With another equation for the EOQ with finite production the following results could be reached. Neglecting the setup time of one week, the finite production could be calculated for a year with 52 weeks * 5 working days * 8 hours * 60 minutes = 124,800 minutes and with a lead time of 56 minutes this resulted in about 2,229 items a year.

With the EOQ formula for finite production (2) it follows:

$$P = 2,229$$
$$Q = \sqrt{\frac{2 \cdot 50 \cdot 653}{3.5}} \cdot \sqrt{\frac{2,229}{2,229 - 653}} \approx 162$$

Both calculations left the problem that the production was not immediate, i.e., there was a lead time until a reorder from the supplier reached the inventory of the retailer. The demand that occurred during this lead time would cause huge stockouts, if there were not any safety stock. So it was necessary to use a reorder quantity model with a reorder point, that was higher than 0.

In this case it could be assumed that the lead time was stable. The ERP system had an entry with an estimate that the lead time took 80 day and one of the orders to the supplier had 590 items. Assuming that this was the lead time in working days for the order of 590 items, the following estimate resulted:

80 days * 8 hours * 60 min/ 590 = 65 minutes.

Another order from the historical data showed a much better situation with 265 items in 40 days that gave a lower lead time per item:

40 days * 8 hours * 60 min / 265 = 32 minutes

But the historical data showed that the delivery was split into several parts and that the first items took longer than the subsequent ones. So there seemed to be a setup time for each order as well. As a consequence for the simulation a rather optimistic estimate for the lead time per item was 56 minutes with a setup time of the complete order of one calendar week, that was 2,400 minutes according to Table 3.

Since the EOQ formula assumed that the delivery of the items was instantaneous, whereas the analysis showed that a setup-time was needed, it was necessary to compensate for this additional time. For this purpose a safety stock was chosen and when the inventory reaches the quantity of this stock, then a reorder was placed.

Assuming the lead time was constant and the demand variable, but basically normally distributed and a service level of 95% should be reached, then the formula (3) could be used. The number of standard deviation Z for a given service level had to be set. The goal was to reach a service level of at least 95%, that means that a 5% probability existed during that lead time demand was higher than the safety stock. For the calculation, the demand that occurred during the lead time had to be added and for 0.05 the number of standard deviation is 1.64 (Waters, 2003).

One disadvantage of the formula was that it did not take into account a lead time that had an added setup time. In fact, the lead time was the time it would need to receive the quantity Q. The lead time in calendar weeks was then:

$$T = (2,400 \, \text{min} + 56 \, \text{min} \cdot Q))/2,400 \, \text{min}$$

for

$$Q = 162$$
$$T = 4.78$$

And for the reorder point calculation resulted:

$$ROP = 4.78 \cdot 13 + 1.64 \cdot 23.44 \cdot \sqrt{4.78} = 146$$

and for

$$Q = 137$$
$$T \approx 4.2$$

that gave a reorder point:

$$ROP = 4.2 \cdot 13 + 1.64 \cdot 23.44 \cdot \sqrt{4.2} = 133$$

That is the result of the analytical solution suggested that with a reorder point of 142 a quantity of 162 items should be reordered and with the reorder point at 133, the quantity should be 137.

The data was analysed according to the service level criteria, but only the sum of the unit-time stockouts and the service level according to deliveries from stock could be calculated. It was not possible to find the numbers for the reorder point or quantity that had been used in the real world scenario, obviously the system was more or less based on ad-hoc decisions.

Since the experiments of the simulation were based on 104 calendar weeks, the analysis divided the historical data of four years in two-year samples with 104 calendar weeks each.

Table 4   Historical stockouts and service level

	Unit-time of stockouts	*Service level*
Year 1 and 2	$2.22 * 10^6$	65%
Year 3 and 4	$1.12 * 10^7$	53%

7   Analysis results

The analysis with cross-sample (n samples 1 result) is summarised in Table 5 that shows the results for the quantity reorder system with (S) and without splitting (N).

Table 5   Results for experiment N and S

	N	*S*
Reorder quantity	94	344
Reorder point	470	394
Minimum mean unit-time	541,436	341,373
Median	208,848	125,328
Standard error	87,310	58,043
Cu	712,565	455,136
Cl	370,308	227,610
Maximum	$\approx 5.99256 * 10^6$	$\approx 4.13309 * 10^6$
Values over *Cu*	21%	18%
No. results with service level 95%	52%	58%

The results show that the service level got worse in both cases with only 52% and 58%, respectively. The situation did not become much better with splitting the deliveries of the supplier, although there was a small improvement, too many results are off from a 95% service level. But it was already clear that the parameters chosen for the simulation did not allow for any better values, because these were the minimum costs calculated for the delays.

Obviously it was not possible to find a solution, that could solely rely on the reorder quantities and the reorder point, i.e., safety stock, in order to guarantee a sufficient service level. Since it was not satisfactory to have either a huge number of delayed items or delay times and at the same time the service level was much too low, the next step examined a crushing of the lead times. The original estimate for the lead time per item was 58 minutes and in the simulation it was now set down to 26 minutes, that was half the original value.

This time for the analysis only the minimum mean unit-time over all samples was taken into the account and the results are summarised in Table 6.

Table 6 Results of experiment NcL and ScL

	NcL	*ScL*
Reorder quantity	92	344
Reoder point	221	213
Minimum mean unit-time	196,343	123,215
Median	69,252	56,322
Standard error	33,609	17,355
Cu	262,217	157,232
Cl	130,470	89,199
Maximum	$\approx 2.37459 * 10^6$	982,656
Values over *Cu*	20%	24%
No. results with service level 95%	71%	77%

It is interesting to see that the change only affected the reorder points, whereas the quantities basically stayed the same. At first it seemed as if the split solution was not much superior, but a comparison of the distributions of the value showed the contrary.

Figure 4 Unit-time values for experiment NcL with half the lead-time (see online version for colours)
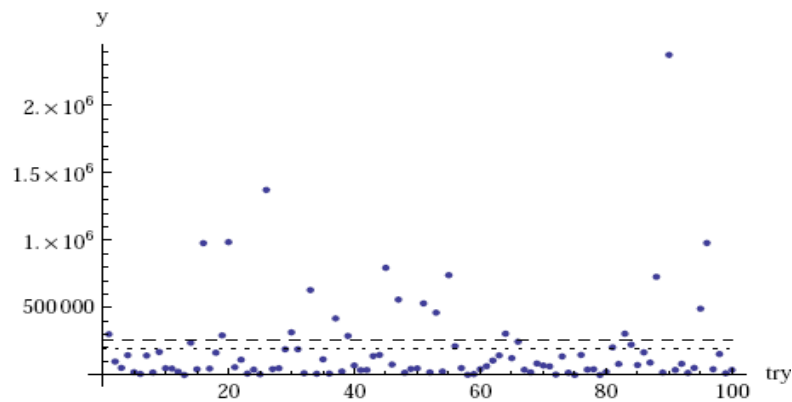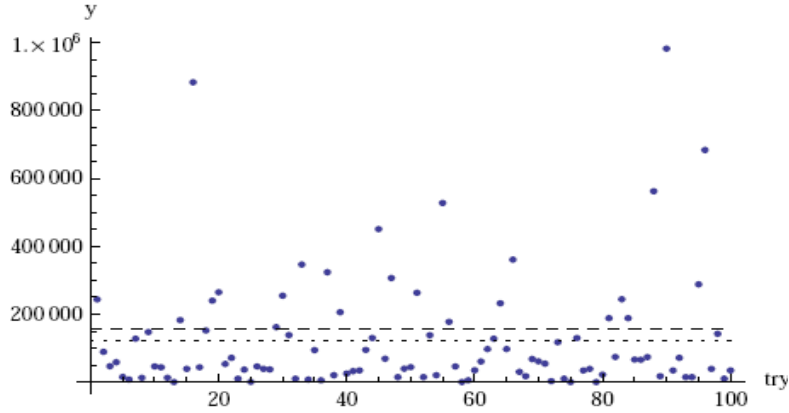
Figure 5   Unit-time values for experiment ScL, split orders, and half the lead-time (see online
version for colours)

The outliers in the split order system of experiment ScL deviated not so much from the
mean value as in the normal case of experiment NcL. Therefore, a system with split
supplier's deliveries was preferable over the other one.

A further experiment ScLS tested a system with half the setup time, that showed the
lowest penalty values on average and thus costs for the stockout in unit-time and the most
solutions that guaranteed a service level of 95%, although there was still a considerable
amount of solutions deviating from this goal.

Another question was whether the values for the reorder quantity and point as well as
for the service level were stable enough. For this reason two more experiments with the
ScLS setup were run, but with other randomly generated demand datasets. Again
100 demand data samples were used so that the results were comparable with the other
experiments. The control experiments have the numbers ScLS1 and ScLS2.

Table 7   Result of experiment ScLS checked against ScLS1 and ScLS2

	ScLS	*ScLS1*	*ScLS2*
Reorder quantity	199	199	199
Reorder point	114	114	114
Mean unit-time	55,515	288,965	282,502
Median	14,280	51,032	39,792
Standard error	11,131	45,067	532,034
Cu	77,331	377,297	386,782
Cl	33,699	200,633	178,222
Maximum	673,600	$2.82248 * 10^6$	$2.25306 * 10^6$
Values greater *Cu*	19%	28%	23%
No. results with service level 95%	84%	60%	64%

Obviously, the reorder point 332 and quantity 99 calculated for the demand data samples of experiment ScLS2 show the best of all results, when compared to the demand data samples of all experiments. The minimum mean stockout penalty values stay within the 95% confidence interval of ScLS2 and at the same time the values for ScLS and ScLS1 even stay within the confidence bounds of their own best mean stockout penalty. Another check for the quality of the solution is the number of results that reach a service level of 95%, that is the best for the reorder quantity and point indicated by ScLS2.

Table 8 Result of experiment ScLS2 checked against ScLS and ScLS1

	SCLS	*SCLS1*	*SCLS2*
Reorder quantity	99	99	99
Reorder point	332	332	332
Mean unit-time	57,252	67,935	76,467
Median	13,496	11,984	7,600
Standard error	10,536	14,809	26,118
Cu	77,902	96,959	127,658
Cl	36,602	38,910	25,276
Maximum	59,1216	$1.1376 * 10^6$	$2.02651 * 10^6$
Values greater *Cu*	22%	18%	11%
No. results with service level 95%	86%	87%	89%

Table 9 Results with EOQ and finite production values

Reorder quantity	137
Reorder point	133
Mean unit-time	$1.80326 * 10^6$
Median	744,188
Standard error	248,470
Cu	$2.29026 * 10^6$
Cl	$1.31626 * 10^6$
Maximum	$1.26967 * 10^7$
Values greater *Cu*	22%
No. results with service level 95%	21%

Table 10 Result with EOQ values

Reorder quantity	162
Reorder point	146
Mean unit-time	$1.58422 * 10^6$
Median	729,304
Standard error	217,493
Cu	$2.0105 * 10^6$
Cl	$1.15793 * 10^6$
Maximum	$1.22473 * 10^7$
Values greater *Cu*	23%
No. results with service level 95%	28%

For a comparison, the situation when the naïve approach of using the analytical formula, without further improvements in the supply chain was chosen. For this purpose, the simulation results from experiment *N* for the analytically calculated reorder points and quantities were taken. In this case, the mean unit-time was not the minimum of all solutions of course.

8   Conclusions

The goal of this research was to analyse the supply chain and the service level for products with an uncertain demand. For this purpose a model of the supply chain was set up and implemented with a simulation based on object-oriented software agents. The service level was evaluated as a stockout penalty. The simulations showed that the situation of the analysed supply chain could be improved dramatically.

One challenge of the research was finding a valid solution for the reorder point and quantity from the simulation and the case study shows that the naïve approach of running some simulations and then taking the result with the optimum stockout penalties from these and determine the reorder point and quantity from it would fail. This is because for another pseudo random demand dataset this solution would not give the optimum stockout penalty. Instead an approach that included all results and took the minimum of the mean stockout penalties over all samples hinted towards a set of reorder point and quantity that gave almost the same stockout penalty when using different demand datasets.

Furthermore, this paper shows that object-oriented simulations are a very good tool to investigate how several parts of a supply chain work together and help to detect weaknesses that are not obvious in an analytical approach. In the case study simulating the supply chain was in fact a valid approach to the original problem, because it revealed that the weakness were the lead times that prevented the system to reach an acceptable service level with an reorder-quantity inventory management system. Furthermore, it allowed to examine how the service level improved with shortened lead times. Another important aspect was that the minimum average stockout penalty over all samples gave a good metric for assessing the costs to be expected.

When comparing the values of the historical data with that of the analytical calculations and the values of the simulation, it shows that the analysed supply chain had the worst performance of all and this applies even though the managers had chosen the order splitting heuristic that should reduce the problem of the demand during lead time. Nevertheless, the service level even declined from 65% to 53% and the calculated stockout penalty rose significantly from $2.22 * 10^6$ to $1.12 * 10^7$.

Compared to this the simulation results with the analytically calculated EOQ values are at least as good as the values from the historical data and this applies without implementing any special heuristics of order splitting, in fact it is really the original model of reordering based on a reorder point with a reorder quantity delivered as a whole. There seems not to be a great advantage of the EOQ with finite production over the EOQ calculation. In this case the conclusion is that the business would have done better with the simple EOQ-based model instead of the mere heuristic approach, even though the EOQ model is not really meant for dealing with stochastic demand situations.

Of course this assumes an ideal situation, because the lead times of the supplier must be rather stable and it is not clear whether the supplier would be able to deliver smaller

quantities on a reliable schedule. Nevertheless if that goal can be reached, then a more rigorous and systematic approach towards the reorder-quantity model for the inventory management pays off with less stockout penalties.

With a more systematic approach the heuristic of order splitting begins to show improved performance as well. This is obvious from the results where with the split order model the stockout penalties declined and the number of solutions with a 95% service level improved. In the real world situation there seemed to be no idea what the right reorder points and quantities would be and the guesses made obviously must have been far off.

At this point the simple Monte Carlo simulation helped to investigate the performance of the supply chain. But first it shows that it is not possible to find an optimum solution that holds for several samples of the pseudo-random demand data. The stability of calculable costs is preferred over the far too risky bet on hitting the optimal solution by chance. Nevertheless the optimum solutions for each sample give an important hint towards the performance of the supply chain, because even with optimum values only 63% or 76% of the solution fulfilled a service level of 95%. When so many optimal solutions lie outside the desired service level, then this is a clear indication that the supply chain itself has a decisive problem, and with the given parameters only the lead time remained.

The research has some limitations. For instance, the demand investigated has the characteristic that is not seasonal, has low variability and is of low volume. Lead time per item is assumed to be constant in the study due to the low variability of the lead time in practice. Obsolescence cost was neglected as it is not important for the product in this case. The analysed network cards are not consumer products, but highly specialised for industrial real-time networks. They are used in installations and devices which are used for decades and not only years (e.g., example power plants or large offset printing machines). Further, research would be required to take into account different environments (e.g., JIT) that require high variable demands and lead times.

It was showed that the split order system was in fact superior, when it was implemented with mathematical rigor and systematically applied in a supply chain. But only with improving, that means lowering, the lead times it was possible to get closer to the goal of a 95% service level. The analytical calculation with the EOQ formula (1) and (2), even when combined with a safety stock as rendered by formula (3) actually give a result, but they do not reveal that the system in most cases will not be able to reach the assumed goal of a service level of 95% and there is no hint towards the problem that the lead times are actually too long.

With a simulation of the supply chain it is possible to reveal the problem of the lead times and by manipulating the lead times themselves it shows how the performance of the system increases. Even if the results for the reorder point and quantity are not absolutely clear, the simulation is able to show the tendency how the supply chain as a whole can improve, when certain parameters are changed and it revealed the impossibility to reach the goal of the service level in the original setup.

The research showed that an object-oriented simulation can be very effective and an excellent alternative to traditional simulation as it offers the ability to reuse objects and supports the creation of library classes that can be used to develop more sophisticated simulations.

References

Anigbogu, S.O., Francisca, O.O. and Usman, K. (2011) 'An intelligent model for sales and inventory management', *Indian Journal of Computer Science and Engineering*, Vol. 2, No. 5, pp.785–791.

Apache.org (2008) *Commons Math 1.2 API: EmpricalDistributionImpl*, available at http://commons.apache.org/math/api-1.2/org/apache/commons/math/random/ EmpiricalDistributionImpl.html (accessed on 27 September 2008).

Buxey, G. (2006) 'Reconstructing inventory management theory', *International Journal of Operations & Production Management*, Vol. 26, No. 9, pp.996–1012, Emerald, available at http://dx.doi.org/10.1108/01443570610682607 (accessed on 10 May 2008).

Chen, I.J. and Paulraj, A. (2004) 'Understanding supply chain management: critical research and a theoretical framework', *International Journal of Production Research*, Vol. 42, No. 1, pp.131–163, Informaworld, available at http://dx.doi.org/10.1080/00207540310001602865 (accessed on 9 February 2008).

Cigolinia, R., Peroa, M. and Rossib, T. (2011) 'An object-oriented simulation meta-model to analyze supply chain performance', *International Journal of Production Research*, Vol. 49, No. 19, pp.5917–5941.

De Sensi, G., Longo, F. and Mirabelli, G. (2007) 'Inventory policies analysis under demand patterns and lead time constraints in a real supply chain', *International Journal of Production Research*, Vol. 99999, No. 1, pp.1–20, Informaword, available at http:dx.doi.org/10.1080/ 00207540701528776 (accessed on 31 May 2008).

Karim, M., Saad, M.F. and Haque, M. (2011) 'Development of a prospective web-based inventory system for management of lab facilities', *Journal of Emerging Trends in Engineering and Applied Sciences*, Vol. 2, No. 1, pp.35–42.

Kobbacy, K.A.H. and Liang, Y. (1999) 'Towards the development of an intelligent inventory management system', *Integrated Manufacturing Systems*, Vol. 10, No. 6, pp.354–366, Emerald, available at http://dx.doi.org/10.1108/09576069910293022 (accessed on 10 May 2008).

Landau, D. (2000) *Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, Port Chester, NY, USA, ebrary, available at http://site.ebrary.com.ezproxy.liv.ac.uk/lib/ liverpool/Top?%3Fnosr=1&id=2000919 (accessed on 21 October 2008).

Silverman, B.W. (1986) *Density Estimation for Statistics and Data Analysis*, available at http://ned.ipac.caltech.edu/level5/March02/Silverman/paper.pdf (accessed on 6 October 2008).

Spedding, T.A. and Chan, K.K. (2000) 'Forecasting demand and inventory management using Bayesian time series', *Integrated Manufacturing Systems*, Vol. 11, No. 5, pp.331–339, Emerald, available at http://dx.doi.org/10.1108/09576060010335609 (accessed on 10 May 2008).

Stefanovic, S. et al. (2008) 'Methodology for modeling and analysis of supply networks', *Journal of Intelligent Manufacturing*, Vol. 19, No. 4, pp.485–503, Springer, available at http://dx.doi. org/10.1007/s10845-008-0098-0 (accessed on 3 October 2008).

Tarim, S.A. and Smith, B.M. (2008) 'Constraint programming for computing non-stationary (R, S) inventory policies', *European Journal of Operational Research*, Vol. 189, pp.1004–1021, ScienceDirect, available at http://dx.doi.org/10.1016/j.ejor.2006.11.048 (accessed on 31 May 2008).

van der Zee, D.J. and van der Vorst, J.G.A.J. (2005) 'A modeling framework for supply chain simulation: opportunities for improved decision making', *Decision Sciences*, Vol. 36, No. 1, pp.65–95, Blackwell Synergy, http://dx.doi.org/10.1111/j.1540-5915.2005.00066.x (accessed on 8 February 2008).

Verma, A.K. (2006) 'Improving agility of supply chains using base stock model and computer based simulations', *International Journal of Physical Distribution & Logistics Management*, Vol. 36, No. 6, pp.445–454, available at http://dx.doi.org/10.1108/09600030610677393 (accessed on 31 May 2008).

Viswanadham, N. et al. (2000) 'Performance analysis and design of supply chains: a Petri net approach', *The Journal of the Operational Research Society*, Vol. 51, No. 10, pp.1158–1169, JSTOR, available at http://links.jstor.org/sici?sici=0160-5682%28200010%2951%3A10%3C1158%3APAADOS%3E2.0.CO%3B2-8 (accessed on 8 February 2008).

Wadhwa, S. et al. (2008) 'Framework for flexibility in dynamic supply chain management', *International Journal of Production Research*, Vol. 46, No. 6, pp.1373–1404, Informaworld, available at http://dx.doi.org/10.1080/00207540600570432 (accessed on 9 February 2008).

Wallin, C., Rungtusanatham, M.J. and Rabinovich, E. (2006) 'What is the 'right' inventory management approach for a purchased item?', *International Journal of Operations & Production Management*, Vol. 26, No. 1, pp.50–68, Emerald, available at http://dx.doi.org/10.1108/01443570610637012 (accessed on 10 May 2008).

Waters, D. (2003) *Inventory Control and Management*, 2nd ed., Wiley, Chichester.