ANALYSING DIFFERENT ORDERING POLICIES IN A SERIES SUPPLY CHAIN BY USING COLOURED PETRI NETS

C. I. Papanagnou and G. D. Halikias
Control Engineering Research Centre
City University, EC1V 0HB, London, UK
Email: {c.papanagnou, g.halikias}@city.ac.uk

Abstract— **The flow of products and information within supply chain networks is an important consideration for practitioners. Simulation tools provide an efficient approach for analysing and validating dynamic systems such as supply chains, where distorted information and poor product management often lead to uncertainty and to instability phenomena. This paper considers the case of a series decentralised supply chain model using Timed Coloured Petri nets (or Timed CP-nets) and analyses the impact of various continuous inventory policies and known forecasting methods followed by supply chain participants. CPN Tools [12] are used for the design of decision-making processes and simulation results are presented to highlight the main issues arising in real systems and to provide insights for future work on modelling and simulation of supply chains.**

Keywords— **Supply chain management, Hierarchical coloured Petri Nets, Inventory control, Forecasting**

I. Introduction

Decentralised supply chains are based on local information and are characterised by lack of coordination between supply chain participants (nodes). The main framework on decentralised supply chains, where decisions are taken locally, is focused on planning and stability issues and the better management of material flows at each node. Decisions are usually associated with different types of ordering policies specifying the amount of orders placed to the upstream node. The primary objective of supply chain management is the integration of products and information flow both upstream and downstream the supply chain by avoiding excessive fluctuations on inventories.

The work presented in this paper discusses the main effects of certain aspects of different inventory policies on the stability and performance of serial multi-node supply chains. In contrast to more traditional inventory-replenishment policies commonly used for supply chain control (e.g., (S, s) or min-max policies), continuous policies and forecasting techniques have only recently been proposed, apparently inspired from the area of classical process control engineering [7], [1], [10]. Their main characteristic is that orders take place continuously, rather than being triggered by specific events (e.g., when inventory falls below a certain target level).

Relative stability in supply chain dynamics is often quantified via the concept of the "bullwhip effect" (demand amplification). The bullwhip effect is a well known instability phenomenon in supply chains, related to increased volatility in demand profiles in the upstream nodes of the chain [5]. This may limit significantly the smooth operation of the chain and result in production planning costs, inventory costs, poor customer service, etc.

Various computer simulation tools have been proposed recently for the analysis of supply chain performance [11]. Most of these tools model supply chains as discrete event systems and examine their behaviour following alternative methodologies. Typically, simulation tools can be used for quantitative analysis (measurement/prediction of variables) or even quantitative analysis (evaluation of reciprocal effects between individual processes).

This paper describes techniques for modelling and simulating supply chains systems via Hierarchical Coloured Timed Petri Nets (HCTPN). The use of Petri nets has been recently proposed in supply chain literature. Mevius and Pibernik [9] present a special type of high-level Petri nets (XML-nets) to introduce an integral approach to supply chain process management. Elmahi et al. [2] propose a Petri net model based on max plus algebra to control supply chains. Landeghem and Bobeanu [4] present a method for modelling supply chains via Petri nets by using the well known example of the Beer Game, while Liu et al. [6] develop a similar approach for modelling event relationships in supply chains.

Makajić-Nikolić et al. [8] use Coloured Petri Nets to study the performance of a series supply chain by means of CPN-Tools. A HCTPN model has been constructed to study the bullwhip effect in decentralized supply chains where individual nodes use aggressive ordering (AO) based on deterministic customer demand patterns. In this paper we extend this simulation model by comparing various scenarios including deterministic end-customer demand profiles in batches and stochastic profiles which are normally distributed. We are especially interested in characterizing demand-amplification between consecutive nodes of the chain. We also analyze the impact of standard forecasting methods such as moving average (MA) and exponential smoothing (ES) on the performance of the system.

II. Description of the supply chain model

We consider a series five-node supply chain consisting of a Manufacturer, Distributor, Supplier, Retailer, and an end Customer site. We assume that there is a single participant at each node. The main characteristics and dynamics of the series supply chain is based on the model

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

presented in [10].We further assume that each participant takes decisions locally and places an amount of orders to the upstream level following an ordering policy based on a forecasting technique. Dispatch of products to downstream levels also depends on certain rules and conditions (i.e. availability). We assume that there is no delay for information transmission and processing between participants, in contrast with the delivery of finished products, where a lead-time delay is associated with each dispatching process.

All these processes can be implemented if we perceive that each action such as placing an order, delivering a product, or taking a decision regarding inventory levels is a discrete event with dynamic properties. Since the number of reachable states in a supply chain is typically very large we need to describe clearly all activities associated with events. This description also helps the user to understand the structure, rules and functions used in the Petri network.

Each time an order is placed on a participant (node), it is first checked whether the amount of products held in stock is sufficient to fulfill it. We assume that there is an initial stock held in all four nodes of the chain. Incoming orders are always served directly from the local inventory stock, which is the sum of the current stock and the amount of products received from the upstream level. In the case of aggressive ordering (AO) [8], if the inventory stores enough products the same order is placed at the upstream node to maintain inventory at the target level. If the stock level is insufficient to fulfill the full order, then the amount of orders placed is set to the amount of products that has not been delivered to the downstream node plus an amount sufficient to fulfil the next order (backlogged orders). When a forecasting technique is followed, the last amount is specified by the forecast updates.

The amount of products delivered to the downstream node also need to be defined. When an order is received, it is first checked if the inventory stores the needed amount of products. If this amount is sufficient, the total amount is dispatched. Otherwise, the manager can either wait for the needed amount to be received and then delivers the requested amount in full, or dispatches the incomplete order immediately, and the overdue additional quantity later, when it becomes available. In this paper both cases are considered, although shipments of incomplete orders are more realistic in supply chains with delivery of strongly standardized products.

The demand behaviour of the end-customer is modelled via a normal distribution with a given variance and mean. Thus, adjustment effects can be observed and examined with permanent changes of demand. On the other hand adjustment-conditioned deviations from the normal distribution can be implemented very easily and easily recognized. This helps the modeler to analyze the effect of demand fluctuations at all nodes of the supply chain, and to monitor better the inventory levels after a single event has occurred. In addition, the model permits additional types of demand processes (i.e. seasonal fluctuations) by modification of the input data. In our model we assume that customer demand can follow orders in batches. Determining the policies at all four stages of the supply chain which can lead to the bullwhip effect is straightforward and we use metrics and graphical representation of backlogged orders to illustrate this phenomenon.

III. Description of the Hierarchical Coloured Petri Net

A. Prime page "Supply chain"

Figure 1 shows the HCPN model of the four-level supply chain. This abstract single prime page, called *Supply chain*, illustrates the highest network level and provides an overview of the entire supply chain network. (Note that the name of each page is displayed on the top left side). Figure 1 also shows how the HCPN has been hierarchically structured into four modules (subnets): The *Manufacturer (M)*, the *Distributor (S1)*, the (intermediate) *Supplier (S2)*, and the *Retailer (S3)*. The subnets of the model are also referred to as "pages", while each submodel node represents a page of the HCPN model. *Supply chain* page has four transitions which all are substitution transitions. The Customer site is represented in the prime page by two different socket places: *Cg* which is associated with the goods received by customers and *Cd* which represents customer demand. *Cg* and *Cd* are the model output and input, respectively. There are also four socket places surrounding all substitution transitions. For instance, page *Supplier* has the following socket places as it can been inferred by Figure 1: *23g* (goods dispatched to *Retailer*), *12g* (goods delivered by *Distributor*), *32d* (demand order received from *Retailer*), and *21d* (demand order placed to Distributor). The hierarchical CP-net model is common for all 3 different ordering policies (aggressive ordering (A0), moving average (MA) and exponential smoothing (ES) forecasting techniques) considered in this paper.

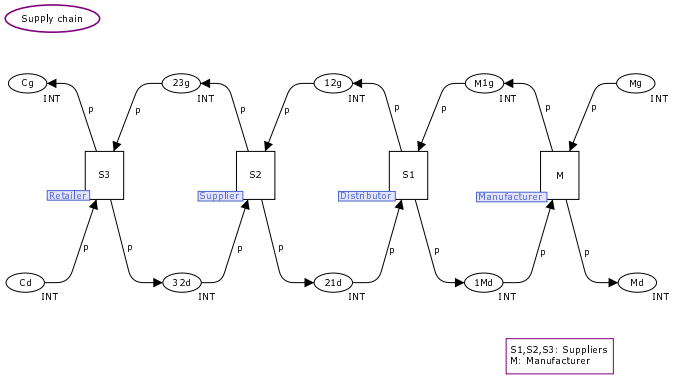


Fig. 1. HCPN describes an overview of supply chain with four different nodes

The three modules *Retailer, Supplier and Distributor*, are structurally identical for each case and their description is based on the *Retailer* module described later, while *Manufacturer* has a different structure which is described in detail. Tokens in the HCPN model are of type "integer" and are associated with the amount of orders and product flow within the supply chain.

Figure 2 depicts the *Retailer* subnet.

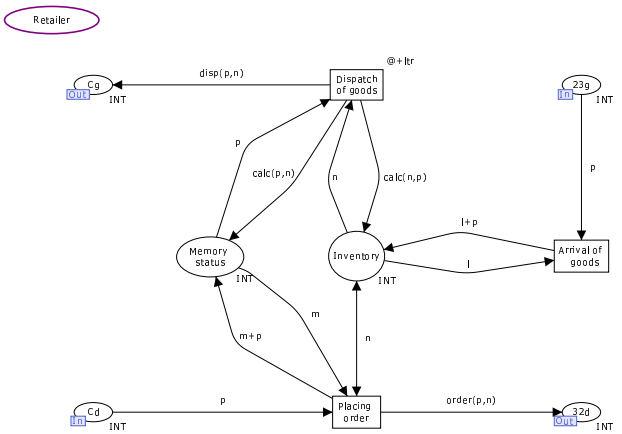


Fig. 2. The subnet Retailer

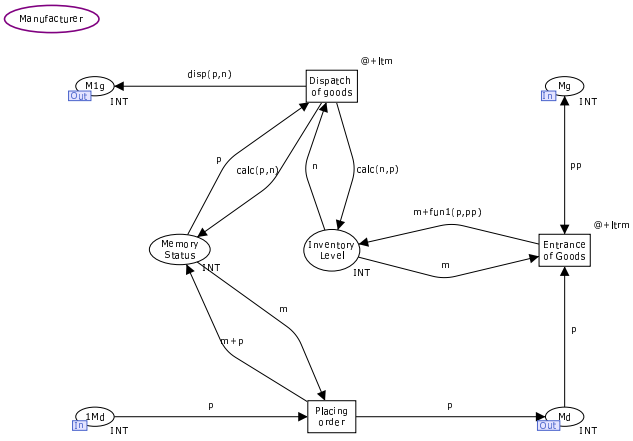


Fig. 3. The subnet Manufacturer

B. Sub-pages Retailer and Manufacturer

B.1 Aggressive ordering (AO)

Figure 2 shows the subnet of the module *Retailer* containing socket places *Cg*, *Cd*, *32d*, *23g*, places *Inventory* and *Memory status*, and three transitions *Dispatch of goods*, *Arrival of goods* and *Placing order*. Transition *Placing order* examines whether the existing stock is sufficient for the complete satisfaction of the quantity requested by the customer. If this is the case, the appropriate order quantity is placed as order to the upstream node (Supplier) and a copy of the quantity is stored in place *Memory status* which plays the role of a buffer. If the existing stock held by the Retailer for a complete supply of goods is not sufficient, then the Retailer orders the amount which is accumulated between the current stock and the last order plus the amount of the last order. This typed of ordering policy is based on the retailer's estimate that the next order placed by the Customer will follow the last demand pattern.

In transition *Dispatch of goods* the ordered products are dispatched to the downstream node (Customer), after being controlled by places *Memory status* and *Inventory*. In the case where the stock is sufficient to fulfil the last order, the Retailer dispatches the whole amount of order. If current stock is less than the order stored in place *Memory status* then the Retailer dispatches the full stock. Transition *Arrival of goods* is associated with goods reception which are later stored on the inventory (warehouse). It is assumed that there is no delay in this process (i.e. goods received by the upstream node are delivered immediately to the warehouse). We also assume that there is no delay on control sequences taking place before the dispatch of goods to the downstream node and on decisions related to the amount of products to be ordered. This assumption is based on the fact that decision-making and dispatch of goods at every node is performed much faster than all other running activities in the supply chain. In contrast, we indicate by *ltr* the lead time between dispatch and delivery of products from Retailer to Customer.

Manufacturer page shown in Figure 3, has exactly the same structure as the Retailer page related to the process of receiving order by an upstream node and dispatching the goods to a downstream node. However, Manufacturers have different policies of receiving and ordering raw materials, since they usually establish contracts monthly or even annually with raw materials suppliers. For the simplification of our model we assume that the Manufacturer receives a settled bulk of raw materials if a current order received by the Distributor exceeds the amount of this quantity. Otherwise, the Manufacturer places exactly the same amount of order for the purposes of maintaining the inventory level. Place *Md* receives the order demand by the Distributor and forwards the amount of order to transition *Entrance of goods* which, in turn, checks if this exceeds the amount of bulk ordering. Place *Mg* models the raw material suppliers' activity and we assume that the time needed for raw materials to be transformed to finished products is *ltrm*. Hence manufacturing site can be considered as a push logistics system where manufacturing process takes place when it is triggered off by downstream demand.

All the inscriptions, functions and variables used for quantitative and qualitative analysis of a model are shown in Figure 4.

```
val ltrm= 1;
val ltm = 4;
val ltd  = 3;
val lts  = 2;
val ltr  = 1;
val s = 8.0;
val m = 100.0;
colset INT = int timed;
type REAL = real;
var q, w, p, pp, m, n, l: INT
normal(m,s) : real;
fun order(i,j) = if(i<=j) then i else 2*1-j;
fun calc(i,j) = if(i>=j) then i-j else 0;
fun disp(i,j) = if(i>=j) then j else i;
fun fun1(i,j) = if(i>=j) then i+j else i;
fun distr(m,s) = floor(normal(m,s));
```

Fig. 4. The declaration box of HCTPN model

To facilitate our analysis, we need to associate the flow of goods and orders by using assigned variables. Moreover,

all activities performed at all stages of the supply chain (such as dispatching of goods, control sequences and order placing) must also be clearly defined via a set of rules. This set can be encapsulated in well-defined functions. CPN Tools for simulation and modelling of CPN use CPN ML, which is obtained by extending Standard ML. All functions and variables used in supply chain model HCPN are written in ML code which extracts all data from the CPN model.

As can be inferred from the declaration box, in order to model the supply chain HCPN with CPN Tools we need to use a standard timed colour set ($INT = int\ timed$). Variable p in all pages represents the amount of orders and goods transferred within the supply chain and is defined as an integer. Variable integer n is used to indicate the amount of stock held at each node. Variable m updates the memory status each time a new order is placed, while l is used in a similar way to update the inventory each time new products are received. In the Manufacturer node, pp is associated with the bulk of raw materials arriving from suppliers Mg each time such a command is issued by the manufacturer. Before carrying out a HCPN simulation, we assume that customer demand Cd follows a normal distribution with mean m and standard deviation s. Both mean and standard deviation are real numbers (100.0 and 8.0, respectively). The normal distribution is implemented in CPN ML code with function $normal(m, s)$. Since CPN Tools can not treat real numbers in the simulation process, we round each random value to an integer by using the function $distr(m, s) = floor(normal(m, s))$.

Function $order(i, j)$ describes the inventory control policy and models the process of placing an order. Variables i and j are input assignments and represent a received order and the inventory, respectively. In case there is sufficient stock, an amount i is sent as order to the upstream node. In cases where current inventory level j is below the received order, then the accumulated value $i - j$ (backorders) plus the estimate of the next order i is placed as order to the upstream node. Function $calc(i, j)$ models the control sequence between memory status and current stock and is used to calculate the balance of inventory and memory status when a new immediate shipment is due to take place. The exact amount of products dispatched to the downstream node is modelled by function $disp(i, j)$.

Function $calc(p, n)$ should not be confused with $calc(n, p)$. Although these two functions operate in a similar way and have the same output assignments, their inputs assignments are different. We have also used variables *ltr, lts, ltd, ltm* as lead times for dispatch of goods by retailer, supplier, distributor, and manufacturer, respectively. Note that we need also to introduce guard functions in transitions *Dispatch of goods* to avoid delivery of zero products.

B.2 Moving average forecasting technique

In the moving average technique, the demand forecast is calculated as the average of the last recorded observations and is given by:

$$\hat{D}_t = \frac{1}{N} \sum_{n=1}^{N} D_{t-n+1} \qquad (1)$$

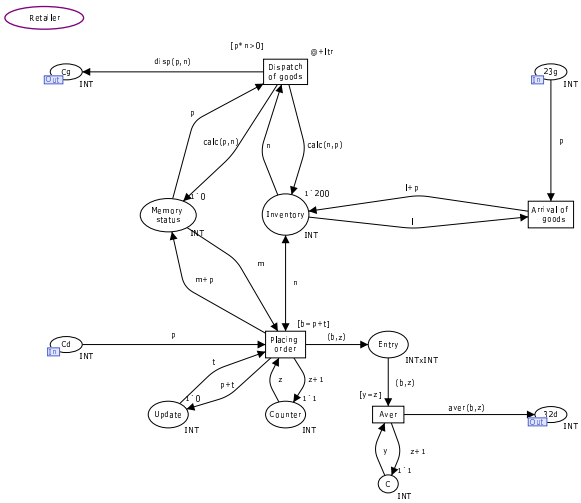


Fig. 5. The subnet *Retailer* for (MA) techniques

where $\hat{D}_t$ is the next demand forecast while D_t, N indicate the amount and number of observations, respectively. Figure 5 shows the corresponding subnet *Retailer* which models the moving average procedure. Recall that system is "driven" by a stochastic end-customer demand profile applied at the end of the chain.

It can be inferred from Figure 5 that we have introduced three new places $Update$, $Counter$ and C and a new transition $Aver$. Place $Update$ is a memory buffer that stores the sum of past observations (e.g., updates the past information) while $Counter$, C and $Aver$ are used to calculate the sum (indicated by the variable b) and the number of past observations (variable z). The guard equality, $y = z$, on transition $Aver$ ensures that data are treated in chronological order. Function $aver(i, j)$ calculates the next demand forecast $\hat{D}_t$ following equation 2. Note that $\hat{D}_t$ must be rounded to an integer since CPN-Tools can not treat real numbers. Hence, two new lines have been added to the declaration box:
fun conv(i) = Real.fromInt((i));
fun aver(i,j) = round(conv(i)/conv(j));

B.3 Exponential smoothing forecasting technique

In cases where we wish to use previous demand forecasts and past observations we can benefit by using the exponential smoothing technique which is given by:

$$\hat{D}_{t+1} = \alpha D_t + (1 - \alpha)\hat{D}_t \qquad , 0 < \alpha < 1 \qquad (2)$$

where α is the smoothing coefficient. This coefficient controls the weight placed on to the most recent data. Figure 6 shows the corresponding subnet *Retailer* which models the exponential smoothing method.

Places Ind and B are used as counter integers and guard equality $z = g$ on transition $estim$, to guarantee, similarly to the moving average model in figure 5, that new data do not overtake older. Place $proest$ forwards the amount of last order observation (variable p) to be processed by transition $estim$. Place A updates the last order (αD_t indicated by variable f) and function $expsmo(p, f)$ calculates

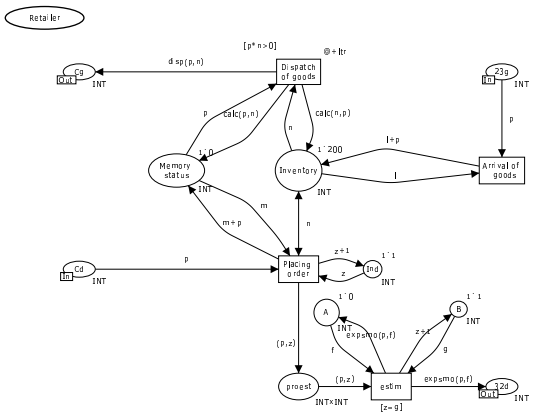


Fig. 6. The subnet *Retailer* for (ES) techniques

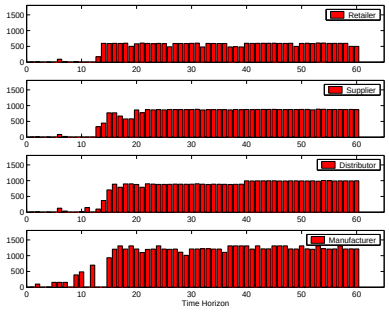
the demand forecasts. Again, all demand forecasts must be rounded to integers. In order to model the smoothing forecast technique we have added the following lines to the declaration box:
val alpha = 0.9;
fun ReaInt(i) = Real.fromInt((i));
fun expsmo(i,j) = round((alpha*ReaInt(i) + (1-alpha)*ReaInt(j)));

IV. Simulation results and performance analysis

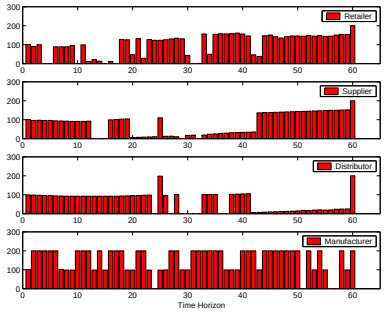
The simulation results show for each ordering and forecasting technique the changes on inventories at each time-period, the corresponding backorders and a customers' satifaction metric. For simplicity we assume constant lead times for the distribution of goods throughout the supply chain. The initial inventory in each node for the (AO) case is set to 100 and 200 for (MA) and (ES). To make our model more realistic, we let participants order in batches but we also consider simulation periods where no orders are placed. Note also that each participant may not dispatch the ordered goods instantly due to administrative delays or laches in the dispatch section. The simulation period has been set to 60 time-steps.

Figure 7 depicts the inventory position of all supply chain participants for all three different cases. Figure 7(a) shows clearly the increase in inventory as we move upstream the supply chain. This observation has also been made in [8]. After a simulation time of sixty periods, the inventory has been heightened by 400%, 780%, 900% and 1100% for Retailer, Supplier, Distributor and Manufacturer, respectively. This inventory augmentation clearly illustrates demand amplification (bullwhip effect).

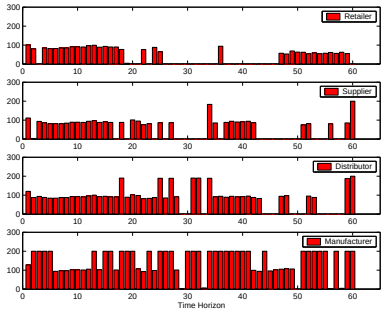
It can be inferred from figure 7(a) that shortages on each participant occurred only on the first 12 time steps. This is mainly caused by batch ordering or when there has been no sufficient inventory to fulfill the downstream demand before aggressive ordering. This can be more clearly seen by considering the Retailer site. The first 12 markings representing customer demand pattern are: $1'98@1 + +1'91@2 + +1'100@3 + +1`98@4 + +1`107@5 + +1`90@7++1`102@8++1`98@8++1`100@10++1`97@11+$



(a) AO



(b) MA

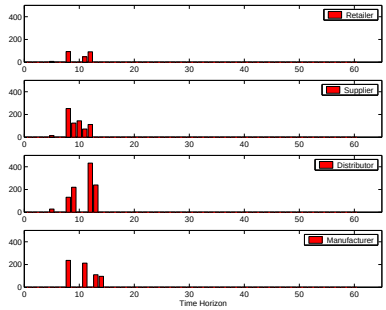


(c) ES

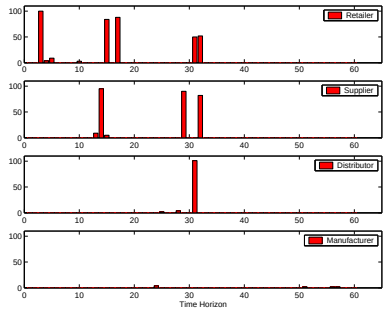
Fig. 7. Inventory levels: (a) AO, (b) MA and (c) ES

$+1`105@12++1`94@12++$ At time $t = 5$ Customer orders 107 products while Retailer's inventory is below 100. Furthermore, at times $t = 8$ and $t = 12$ customer places two different orders. Due to aggressive ordering (AO) adopted by Retailer at these times, inventory at the 13th time step increases significantly.

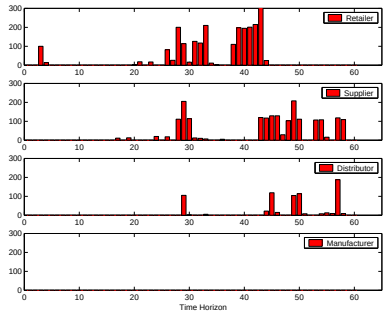
The shortages on inventories depicted in figures 7(b) and 7(c) demonstrate the impact of batch ordering in forecasting techniques. In the MA case, batch ordering followed by customer takes place at periods represented by markings: $1`104@3 + +1`107@3$, $1`92@12 + +1`99@12$. Delays on dispatching the goods to Retailer by Supplier on consecutive simulation periods 31 and 32, and to Supplier by Distributor at periods 13,14,15,29 and 32 also cause shortages in inventories. Similarly, zero inventories appear in the Dis-



(a) AO



(b) MA



(c) ES

Fig. 8. Backorders (a) AO, (b) MA, and (c) ES

tributor's and Manufacturer's sites, caused by delays on dispatches and batch ordering. As shown in figure 7(c) for the ES technique with α=0.9, there are more periods where participants encounter inventory shortages in comparison to MA technique. This means that batch ordering and delays on goods dispatches using the ES method tend to have more impact on inadequacy of supplies. Note also that ES produces slightly more fluctuations on inventories than the MA technique, although inventory levels on MA are very low for long periods (e.g., Distributor's inventory for the last 20 periods is below 15 and Retailer's inventory between the 26th and 40th period is below 20).

Figures 8(a), 8(b) and 8(c) depict the backorders in supply chain participants for AO, MA and ES techniques, respectively. It can be inferred that due to the more inventory shortages when participants follow ES forecasting techniques the number of backorders is also larger with this method. Note also that backorders for MA and AO appear mainly on the first half of the simulation period in contrast to ES where they appear on the second half. Figure 8 illustrates that backorders in smoothing forecasts take place at the downstream part of the supply chain in contrast with AO where inventory shortages arise upstream. As expected from the observed backorder levels, customer satisfaction with ES is lower compared to the other two techniques.

V. Acknowledgments

C.I. Papanagnou would like to thank Greek State Scholarships Foundation (I.K.Y.) for the financial support of his research.

VI. Conclusions

In this paper we presented a simulation framework for analysing a series supply chain system. With the aid of a simulation model based on HCTPN we analysed three different ordering policies widely used in supply chain management. Simulation results suggest that MA continuous policies offer advantages for the scenarios analysed. Using this technique we can reduce the bullwhip effect and avoid high backorder levels, while customer satisfaction is increased. However, supply chains are complex dynamic systems with intricate interrelations and cumbersome functions. Therefore, additional modelling work is required to capture all characteristics of real supply chains.

References

[1] J. Dejonckheere, S.M. Disney, M.R. Lambrecht and D.R. Towill. *Measuring and avoiding the bullwhip effect: A control-theoretic approach*, European Journal of Operational Research, 147, (2003) 567-590.

[2] I. Elmahi, O. Grunder and A. Elmoundni *A Petri net approach for the evaluation and command of a supply chain using the max plus algebra*, Proc. of the 2nd IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia, 2002.

[3] K. Jensen *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Vol. 1, 2, 3 Springer-Verlag, 1997.

[4] R. van Landeghem and C. M. Bobeanu *Formal modelling of supply chain: An incremental approach using Petri nets*, Proc. of the 14th European Simulation Symposium, Dresden, Germany, 2002.

[5] H.L. Lee, V. Padmanabhan, S. Whang, *The Bullwhip effect in supply chains*, Ind. Eng. Chem. Res., 40, (2001) 3369-3383.

[6] E. R. Liu, A. Kumar and W. van der Aalst *A formal modeling approach for supply chain event management*, Proc. of the 14th Workshop on Information Technologies and Systems, Washington DC, 2004.

[7] Pin-Ho Lin, D. Shan-Hill Wong, Shi-Shang Jang, Shyan-Shu Shieh and Ji-Zheng Chu. *Controller design and reduction of bullwhip for a model supply chain system using z-transform analysis*, Journal of Process Control, 14 (2004) 487-499.

[8] D. Makajić-Nikolić, B. Panić and Mirko Vujošević *Bullwhip effect and supply chain modelling and analysis using CPN Tools*, Proc. of the 5th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools. Aarhus, Denmark, (2004) 219-234.

[9] M. von Mevius and R. Pibernik, *Process management in supply chains - A new Petri-net based approach*, Proc. of the 37th Hawaii International Conference on System Sciences, 2004

[10] C.I. Papanagnou and G.D. Halikias *A state-space approach for analysing the bullwhip effect in supply chains*, Proc. of the 5th International Conference on Technology and Automation, Thessaloniki, Greece, (2005) 79-84.

[11] S. Terzi and S. Cavalieri *Simulation in the supply chain context: a survey*, Computers in industry 53, 1, (2004) 3-16.

[12] http://wiki.daimi.au.dk/cpntools/cpntools.wiki