DASCh: Dynamic Analysis of Supply Chains

H. Van Dyke Parunak*, Robert Savit, Rick L. Riolo**, Steven J. Clark***

*Center for Electronic Commerce
ERIM, Inc.
PO Box 134001
Ann Arbor, MI 48113-4001
vparunak@erim.org

**Program for the Study of Complex Systems
Randall Laboratory
University of Michigan
Ann Arbor, MI 48109-1120
{savit, rlriolo)@umich.edu

0. Executive Summary

Modern manufacturing is moving away from vertically integrated companies that control all aspects of production and distribution, toward networks of independent suppliers and distributors. These supply networks (more commonly called "supply chains") address a number of business needs, including concentration on core competencies and the ability to respond rapidly to unanticipated change. However, they present significant operational challenges. Many of these challenges are driven by the dynamical behavior of the supply chain as its members interact with one another. Data analytic approaches are not effective in understanding these dynamics, because the commercial environment changes too rapidly to permit the collection of consistent data series long enough to support statistical requirements. DASCh takes the approach of constructing and experimenting with an agent-based emulation model of the system that can maintain a given set of conditions as long as desired.

0.1. Previous Work

Three different approaches have been taken to the problem of modeling and analyzing supply chains.

Control theorists model the chain with differential or difference equations and use transform analysis to explore their behavior. This approach is dynamical but relies heavily on linearity assumptions that are not satisfied in most supply networks, for reasons discussed in the problem statement section.

Classical *operations research* approaches include optimization theory, game theory, and statistical analysis. These tools apply to nonlinear as well as linear systems, but often make unrealistic statistical assumptions. In addition, they are not explicitly time-based, and so cannot capture the dynamical characteristics of the system.

Simulation approaches experiment with an executable model of the system. In most cases these models are in support of one or the other of the previous two approaches. Virtually all simulation work to date models the supply chain as a set of differential equations and then integrates these equations over time. DASCh uses agent-based modeling, representing the various components of the supply chain by software agents that emulate their actual behaviors. The DASCh approach is more faithful than equation-based modeling, better supports the increasingly decentralized nature of supply chains and the need to protect proprietary information, and provides a much closer interaction between model and real system.

0.2. Model Structure

The agents currently implemented in DASCh are of three species. *Company agents* represent the different firms that trade with one another in a supply network. They consume inputs from their suppliers and transform them into outputs that they send to their customers. *PPIC agents* model the production planning and inventory control algorithms used by company agents, and currently support a simple MRP model. *Shipping agents* model the delay and uncertainty involved in the movement of both material and information between trading partners. To insure the realism of this model in spite of its simplicity, we recruited a Fortune-100 manufacturing manager as a project advisor. He guided our decisions during model construction and reviewed the results we obtained. In spite of the restricted scope of the model, he found its results of sufficient interest and credibility that he implemented changes in his own operations based on the model, and has observed subsequent performance improvements that he attributes to these changes.

0.3. Theory

One of the promises of DASCh is in analyzing nonlinear systems, which in general do not yield to analytical solution. However, where theoretical analysis is possible, it adds insight and can help direct experimentation. We have developed theoretical treatments for two aspects of the behavior of DASCh. The first predicts the phenomena of amplification and correlation of variance in the order stream even in DASCh's its linear domain. The second describes details of the oscillatory behavior of inventory levels under the imposition of a threshold nonlinearity in site capacity.

0.4. Experimental Results

The experiments described in this report involve a linear supply chain with four company agents (a boundary supplier, a boundary consumer, and two intermediate firms producing a product with neither assembly nor disassembly). The two intermediate company agents each have PPIC agents to convert incoming orders to orders for their inputs, and shipping agents manage all movement of both material and information among company agents.

This simple structure was intended as a starting point. It was expected to yield relatively uninteresting behavior, on which the impact of successive modifications could be studied. In fact, it shows a range of interesting behaviors in terms of the variability in orders and inventories at the various company agents.

- As the demand generated by the top-level consumer propagates to lower levels, its variance increases, so that lower-level suppliers experience much more variability than higher-level ones. This phenomenon is widely discussed in the literature.
- Not as well recognized in the literature is the correlation imposed on an originally uncorrelated series of random orders by the PPIC algorithms in the supply network.
- A single modest change at the top of the supply chain generates disturbances in the order sequences of lower tier suppliers that persist long after the original change.

- Even when top-level demand is constant and bottom-level supply is completely reliable, inventory levels at intermediate sites can generate complex oscillations in inventory levels, including period doubling, as a result of capacity limitations.

The detailed discussion of the experimental results identifies operating parameters that affect these behaviors.

0.5. Summary and Recommendations

The insights from DASCh relate to the four characteristics of dynamical systems described in the original proposal (accessibility, controllability, inertia, and performance), and lead to several recommendations for actual trading practices. The DASCh research may profitably be extended in four directions: quantitative analysis of real manufacturing data guided by the behaviors we observe in the model, further experimentation with the current model, structural extensions to the model to support arbitrary network structures, and adaptive site-level behaviors to compensate for undesirable system-level dynamics.

1. Previous Work

This part reviews previous quantitative research on the behavior of supply chains. It groups the relevant literature according to technical approach, summarizes the behaviors that have been identified, and compares our work with this background.

1.1. General Approaches

Three different approaches have been taken to the study of supply networks.

1. *Control theorists* model the chain with differential or difference equations and use transform analysis to explore their behavior. This approach is dynamical but relies heavily on linearity assumptions that are not satisfied in most supply networks, for reasons discussed in the problem statement section.
2. Classical *operations research* approaches include optimization theory, game theory, and statistical analysis. These tools apply to nonlinear as well as linear systems, but often make statistical assumptions and are not explicitly time-based.
3. *Simulation approaches* experiment with an executable model of the system. In most cases these models are in support of one or the other of the previous two approaches. These in turn are of two broad classes: equation-based modeling (the dominant approach until now) and agent-based modeling (our approach).

1.1.1. Control Theory

This approach originated with [22], and is part of the work for which Simon was awarded the 1978 Nobel Prize in economics. It uses tools developed for the study of differential or difference equations through time, particularly Laplace and Z transforms. Thus it is explicitly a dynamical approach, sensitive to the time-based behavior of a system. However, reliance on transforms makes it most naturally applicable to linear systems. A closely related line of work, discussed under “Simulation” below, also models systems with difference or differential equations, but solves the equations numerically rather than through transforms, and thus is not restricted by linearity assumptions.

[4] develops a Z-transform of a time-averaged ordering rule and shows by composition up the chain that amplification is partly due to adding safety stock on top of safety stock. He recommends dividing the order from the immediate customer into two parts, one reflecting actual changed demand from the end customer, the other reflecting adjustments made by the immediate customer for his own purposes. A given node in the chain should adjust its production only to changes in the end customer’s demand. The recommended mechanism for distinguishing the two components of the order is detailed knowledge of the immediate customer’s ordering policy, from which a producer can derive its own policy. However, disclosure of these policies between companies is problematic. Even if firms are willing to disclose such information, they may well distort their disclosures to manipulate their trading partners.

[27] models demand within a single echelon (level of the supply chain) as average consumption plus fraction of inventory deficit. An elaborate Laplace transform analysis shows that three parameters should be about the same if the system is to settle quickly but without oscillation: time to adjust inventory, demand averaging time, and production delay time. [29] extends this

approach to decompose the Forrester model into cascading echelons and use transfer functions to analyze their behavior, focusing on amplification of order variance. The work of Towill and his team includes simulation as well as transform analysis. The simulation components are summarized below.

1.1.2. OR Analysis

These studies draw on classical OR techniques such as optimization, game theory, and statistical analysis. They do not assume linearity, but do make other strong assumptions about the underlying statistical distributions, and focus on time averages and steady states rather than dynamical behavior.

1.1.2.1. Optimization

[16] studies four possible causes of the amplification of demand variance. Two of these draw on optimization theory.

- Demand signal processing: Formulates the cost minimization problem for a retailer and shows that processing historical demand to forecast future demand in order to minimize cost results in variance in outgoing orders that is strictly larger than sales variance. The variance increases with lead time.
- Price variations: Formulates a retailer's buying policy in the face of fluctuating prices. Unreported optimization computations show that the optimal policy is to let inventory drop during times of high price, and stock up during periods of low price, thus providing an additional source of variation beyond that in the customer's demand.

[30] uses Lagrangian optimization to compare costs under three different mechanisms of operation for three-level chain: optimization of the top level only; optimization across all levels; local optimization at each level. He shows that the third approach gives better results than the first, at a fraction of the computational effort of the second.

1.1.2.2. Game Theory

One of four causes of variation amplification studied in [16] is the "rationing game," in which purchasers competing for a scarce input overstate their needs in order to be sure of getting enough. Analysis of the Nash equilibrium for these competing purchasers shows that the result of such behavior is to increase variation in the orders they place with respect to that in their incoming demand.

1.1.2.3. Statistical Analysis

These studies typically compute various moments of critical quantities, such as demand or inventory levels. They are not bound by linearity assumptions, but they bring in a host of other assumptions, such as the forms of distributions that generate various quantities or the statistical independence of samples from one another. In addition, they deal with averages over time, and so are not truly dynamical.

One of four causes of variation amplification studied in [16] is order batching (combinations of incoming orders from a number of retailers). The analysis statistically derives the variance of the overall order stream from the variances of the individual streams under assumptions of random, positive correlation, and balanced ordering. If σ^2 is the variance of each of the N retailers, this analysis shows that Var(Correlated) ≥ Var(Random) ≥ Var (Balanced) ≥ $N*\sigma^2$.

[25] offers a static analysis of the material costs experience by a manufacturer depending on whether or not suppliers reveal their capacity constraints, and shows that disclosure of capacity constraints benefits the manufacturer and the lower-priced supplier but not necessarily the higher-priced one.

[14] develops models of inventory levels and supplier response time on the input side of the network, based on a tolerance stacking model. These methods yield the standard deviation of actual demand, from which a reorder point can be defined.

[15] derives the relation between base stock level and a target service level, using standard analysis of statistical moments.

[5] looks at demand variance, inventory holding costs, and backorder costs on the distribution side of a supply network, using algebraic analysis and an unspecified simulation model. He explores three variables: whether retailers' orders are aligned or not (cf. [16]); the interval between orders from a given retailer; and the minimum order quantity permitted. Three kinds of action are found to be effective in lowering the variance in the net demand stream issuing from the set of retailers, with different impact on overall supply chain costs:

1. Balancing orders by forcing a fixed interval between successive orders and staggering who can order when lowers costs.
2. Raising the interval between orders tends to raise costs.
3. Raising interval but dropping order quantity so as to keep retailer order frequency constant reduces costs as well as variance.

1.1.3. Simulation and Emulation

These approaches actually execute a model of the system and observe its behavior experimentally. The general name for such an approach is “simulation.” Within simulation, we distinguish two varieties: equation-based modeling and agent-based modeling. In addition of DASCh, we know of only two other examples of agent-based modeling, [26] and [17, 24]. [26] outlines the structure of a supply chain modeling system, with structural and control elements, but reports no results. [17, 24] describe a framework and some simple average results on inventory levels and cycle times, but no dynamical analysis. In addition to the studies collected here, [5] claims numerical confirmation of his theoretical results, but offers no details on structure of the model, platform, or methodology.

Virtually all simulation studies of supply chain dynamics rely on the integration of differential or difference equations, providing an experimental counterpart to the control theory approach. The work of Forrester and his students with these techniques has led to the field of “systems dynamics.” It enjoys an active professional society [6], extensive literature including methodological texts [9, 21], supporting software (including DYNAMO, iThink, Vensim, and PowerSim), and consulting firms that specialize in this approach (e.g., Forrester Consulting, Pugh-Roberts Associates, Decision Dynamics). This modeling approach, which has been applied to a wide range of “soft” policy studies, focuses on “what-if” games with various control variables rather than directly emulating the intrinsic behavior of the elements of the chain.

[9] formulates supply chains as difference equations and then uses Dynamo to sum them numerically. His models lump variables across the entire chain into one system, without maintaining any disciplined division among individual entities.

[23] follows Forrester's lead, but pays more attention to the divisions between entities. He develops a difference equation model of each echelon in the chain, uses it to set upper bounds on optimal behavior, and fits it to the observed behavior of humans in the Beer Game. Focuses on distribution side, but with a nod to raw materials. He observes not only amplification of variance (which many other researchers note), but also (briefly) oscillation and phase lag.

[1] develops a detailed model of the interaction of product makers and the manufacturers of the machine tools that they use, including effects of a step function increase in demand, available workforce, production lead-time for machine tools, and smoother ordering and operating policies, and compares it to empirical observations. The model lumps together the parameters for each sector, representing all product makers by a single instance of the equations for "product maker," and all machine makers by a single instance of the equations for "machine maker."

Towill and his associates support their control theoretic analysis with simulation studies. [28, 33]. A numerical evaluation, based on Towill's model of the echelon, shows the effect of two components of an order: the pass-through of the original demand from one's customer, and the added amount one imposes to manage internal demand variation. The results suggest that these should be passed along the supply chain separately, a conclusion similar to that reached analytically by [4].

1.1.4. Observed Behavior

A brief comment in [23] recognizes three main behaviors in supply chains: amplification of demand variation, oscillation, and phase lags. He relates oscillation to capacity limits in the factory and phase lags to the time needed for information and material movement and processing. His and other studies provide detailed discussion only of the amplification of demand variation from one echelon to the next.

This amplification was observed by [9] and has been highlighted by [3] as the "law of industrial dynamics." The studies surveyed above offer a number of recommendations to fix this problem:

- Balance retailer requests on the supplier [16], [5].
- Track what is in the pipeline [23].
- Eliminate excessive layers (e.g., the distributor) [28].
- Integrate information flow throughout the chain [28]. [4] shows that amplification results when one echelon applies its safety corrections to a lump result from a previous echelon that includes not only original demand but also the previous echelon's safety corrections. Better information flow could avoid this double-counting. Two approaches have been proposed: passing along two lines of orders [33], and having each echelon deduce its ordering policy from algebraic manipulation of the policy of the previous level [4]. Both approaches are subject to gaming behavior when there is competition for scarce resources [16].
- Reduce time delays [28].
- Improve pipeline policy [28].

- Tune parameters of order algorithms [16, 28].
- Increase time between orders, while decreasing minimum order quantity [5].
- Keep time to adjust inventory, production delay, and demand averaging time all about the same [27].

Many of these recommendations founder on the problem of the commons. [16] in particular shows that behaviors that are locally rational for an individual firm may exacerbate bad effects at the system level.

1.2. Where does DASCh fit?

DASCh falls within the "simulation and emulation" approach to supply chain analysis, since we wish to address nonlinearities that are not accessible to analytic control theoretic formulations and dynamical effects that are not visible in traditional OR approaches. Our approach is agent-based modeling (ABM) rather than equation-based modeling (EBM). Based on our preliminary results with such a model [18, 19], we have observed a broader range of supply chain behaviors than has been documented by other researchers, leading to correspondingly richer practical recommendations.

1.2.1. Agent-Based Behavioral Emulation

To our knowledge, the only work on an ABM of a supply network (as opposed to an EBM) is that of [26] and [17, 24], and the focus of those teams has been on the structure of the model and on the kind of average-based analysis typical of other approaches, not dynamical results.

A practitioner is concerned with the underlying *structure* of a model, the naturalness of its *representation* of a system, and the *verisimilitude* of a straightforward representation. This section discusses these considerations with special reference to modeling supply networks. Some of these issues have been discussed by others in the domains of social science [2, 7] and ecology [13, 31] (where ABM's are usually called "Individual-Based Models").

1.2.1.1. Model Structure

The difference in representational focus between ABM and EBM has consequences for how models are modularized. EBM's represent the system as a set of equations that relate observables to one another. The basic unit of the model, the equation, typically relates observables whose values are affected by the actions of multiple individuals, so the natural modularization often crosses boundaries among individuals. ABM's represent the internal behavior of each individual. One agent's behavior may depend on observables generated by other individuals, but does not directly access the representation of those individuals' behaviors, so the natural modularization follows boundaries among individuals.

This fundamental difference in model structure gives ABM a significant advantage in commercial applications such as supply network modeling, in two ways.

1. In an ABM, each firm has its own agent or agents. An agent's internal behaviors are not required to be visible to the rest of the system, so firms can maintain proprietary information about their internal operations. Groups of firms can conduct joint modeling exercises while keeping their individual agents on their own computers, maintaining whatever controls are

needed. Construction of an EBM requires disclosure of the relationships that each firm maintains on observables so that the equations can be formulated and evaluated. Distributed execution of EBM's is not impossible, but does not naturally respect commercially important boundaries among the individuals.

2. In many cases, simulation of a system is part of a larger project whose desired outcome is a control scheme that more or less automatically regulates the behavior of the entire system. The agents in an ABM correspond one-to-one with the individuals (e.g., firms or divisions of firms) in the system being modeled, and their behaviors are analogs of the real behaviors. These two characteristics make agents a natural locus for the application of adaptive techniques that can modify their behaviors as the agents execute, so as to control the emergent behavior of the overall system. The migration from simulation model to adaptive control model is much more straightforward in ABM than in EBM. One can easily imagine a member of a supply network using its simulation agent as the basis for an automated control agent that handles routine interactions with trading partners. It is much less likely that such a firm would submit aspects of its operation to an external "equation manager" that maintains specified relationships among observables from several firms.

More generally, ABM's are better suited to domains where the natural unit of decomposition is the individual rather than the observable or the equation, and where physical distribution of the computation across multiple processors is desirable. EBM's may be better suited to domains where the natural unit of decomposition is the observable or equation rather than the individual.

1.2.1.2. System Representation

The variety of EBM with which we have experimented (ODE's) most naturally represents the process being analyzed as a set of flow rates and levels. ABM most naturally represents the process as a set of behaviors, which may include features difficult to represent as rates and levels, such as step-by-step processes and conditional decisions. ODE's are well-suited to represent purely physical processes. However, business processes are dominated by discrete decision-making. This is only one example of representational advantages of ABM's over EBM's. More generally:

- ABM's are easier to construct. Certain behaviors are difficult to translate into a consistent rate-and-level formalism. PPIC algorithms are an important example. In our attempts to duplicate DASCh results using VenSim®, we were unable to construct a credible PPIC algorithm using the rate-and-level formalism. [32] comments on the complexity of such models, and we have been unable to find an actual example of such a model in the system dynamics literature. Recent enhancements to ithink® reflect such difficulties. The most recent release of this popular system dynamics package includes "black boxes" for specific entities such as conveyors or ovens whose behavior is difficult to represent in a pure rate-and-level system [10]. One suspects that the only realistic way to incorporate complex decision algorithms such as PPIC in system dynamics models will be by implementing such black boxes, thus incorporating elements of ABM in the spirit of [8].
- ABM's make it easier to distinguish physical space from interaction space. In many applications, physical space helps define which individuals can interact with one another. Customer-supplier relationships a century ago were dominated by physical space, leading to the development of regional industries, such as the automotive industry in southeast

Michigan. Advances in telecommunications and transportation enable companies that are physically separate from one another to interact relatively easily, so that automotive suppliers in Michigan now find themselves in competition with suppliers based in Mexico or the Pacific rim. Such examples show that physical space is an increasingly poor surrogate for interaction space in applications such as commerce. ODE methods such as system dynamics have no intrinsic model of space at all. PDE's provide a parsimonious model of physical space, but not of interaction space. ABM's permit the definition of arbitrary topologies for the interaction of agents.

- ABM's offer an additional level of validation. Both ABM's and EBM's can be validated at the system level, by comparing model output with real system behavior. In addition, ABM's can be validated at the individual level, since the behaviors encoded for each agent can be compared with local observations on the actual behavior of the domain individuals. (A balancing consideration is that the code needed to represent an agent's behavior in ABM is often longer and more complex than a typical equation in an EBM, and thus potentially more susceptible to representational error.)
- ABM's support more direct experimentation. Managers playing "what-if" games with the model can think directly in terms of familiar business processes, rather than having to translate them into equations relating observables.
- ABM's are easier to translate back into practice. One purpose of "what-if" experiments with a model is to identify improved business practices that can then be implemented in the company. If the model is expressed and modified directly in terms of behaviors, implementation of its recommendations is simply a matter of transcribing the modified behaviors of the agents into task descriptions for the underlying physical entities in the real world.

1.2.1.3. Verisimilitude

In many domains, ABM's give more realistic results than EBM's, for manageable levels of representational detail. The qualification about level of detail is important. Since PDE's are computationally complete, one can in principle construct a set of PDE's that completely mimics the behavior of any ABM, and thus produce the same results. However, the PDE model may be much too complex for reasonable manipulation and comprehension. EBM's (like system dynamics) based on simpler formalisms than PDE's may yield less realistic results regardless of the level of detail in the representation.

One example in the case of extremely simple agents is the Ising model of ferromagnetic phase transitions in statistical physics. The agent in this model is a single atom in an N-dimensional square lattice of similar agents. Its behavior is to change the orientation of its spin to minimize the energy in its environment. One common and generally useful approach to such systems employs mean field theory, analyzing the behavior of a representative atom under statistical averages over the states of neighboring atoms [20, pp. 430-434]. In some dimensions, this mean field EBM approach may miss the order of the phase transition, predict a phase transition where there is none, or yield an inaccurate temperature for the transition. (In one and two dimensions, the equations defining the Ising model can be solved exactly and analytically without the homogeneity assumptions that lead to the errors of the mean field approach, but such solutions are intractable in higher dimensions.) ABM models that emulate the behavior of individual atoms can be developed for arbitrary dimensions, and are more accurate both qualitatively and quantitatively than the mean field approximation.

In a more complex domain, researchers in the dynamics of traffic networks have achieved more realistic results from traffic models that emulate the behaviors of individual drivers and vehicles, compared with the previous generation of models that simulate traffic as the flow of a fluid through a network [12]. The latter example bears strong similarities to the flow-and-stock approach to supply chain simulation, and encourages us to develop an agent-based approach for this application as well.

Wilson [34] offers a detailed study that compares ABM and EBM using the same system (a predator-prey model). He develops a series of EBM's, each enhancing the previous one to rectify inconsistencies between the ABM and the EBM. The study assumes that the ABM is the more realistic model, and that the EBM is the appropriate locus for making adjustments to bring the two models into agreement. The initial ODE EBM describes reactions between the two species, but representing dispersal through space requires extending it to a set of spatio-temporal integro-differential equations. These equations, modeling both individual characteristics and dispersal using population averages, lead to qualitatively different behaviors than do ABM's. For example, ignoring local variation in dispersal leads to limit cycles rather than the extinction scenarios that dominate ABM's. To correct for these lumped parameter effects, the EBM is interrupted at each iteration of the integration to add a random perturbation to the population parameter at each location and to zero local population levels that fall below specified threshholds.

The disadvantages of EBM in these examples result largely from the use of averages of critical system variables over time and space. They assume homogeneity among individuals, but individuals in real systems are often highly heterogeneous. When the dynamics are nonlinear, local variations from the averages can lead to significant deviations in overall system behavior. In business applications, driven by "if-then" decisions, nonlinearity is the rule. Because ABM's are inherently local, it is natural to let each agent monitor the value of system variables locally, without averaging over time and space and thus without losing the local idiosyncrasies that can determine overall system behavior. The EBM used in our experiments does not use averages over individuals, and so does not suffer from this disadvantage. However, real-world supply networks are much larger. The total number of shipping points in the U.S. automotive industry is on the order of 40,000, and it is difficult to see how a parsimonious EBM of such a system could avoid the use of lumped parameters.

1.2.2. DASCh Preliminary Results

The results described later in this report include the behavior of variation amplification discussed by other researchers, but with new quantitative details. In addition, we offer the first systematic discussion of the generation and persistence of variation. These are specific examples of our distinctive emphasis on understanding the range of overall dynamics of the system rather jumping immediately to detailed analysis of causes for a single system behavior.

2. Model Structure

This part describes the structure of the DASCh model and the behaviors and parameters associated with its various components. The structure of the model and behavior of its individual agents were developed in close consultation with the Manager of Electronic Planning at manufacturing facilities of a Fortune-100 electronics manufacturer to ensure that, though simplified to facilitate initial exploration, they are still realistic and the underlying assumptions are representative of industrial practice.

2.1. The Model

The model represents a supply chain consisting of an OEM, consumer demand for its product, and a supplier of its raw materials. Only one product is modeled, and it is manufactured from only one raw material. The OEM actually has several manufacturing sites, although the single product is manufactured at only one of them, so only that one manufacturing site is modeled. The OEM has a separate centralized shipping site, making the flow of goods from supplier to OEM manufacturing site to OEM shipping site to consumer. The agents representing these four entities are called Supplier-4, Site-3, Site-2, and Consumer-1, respectively. Figure 2.1 shows the interconnection of these agents. The DASCh software permits construction of supply chains of any length, but the experiments reported in this document use this simple four-level chain.

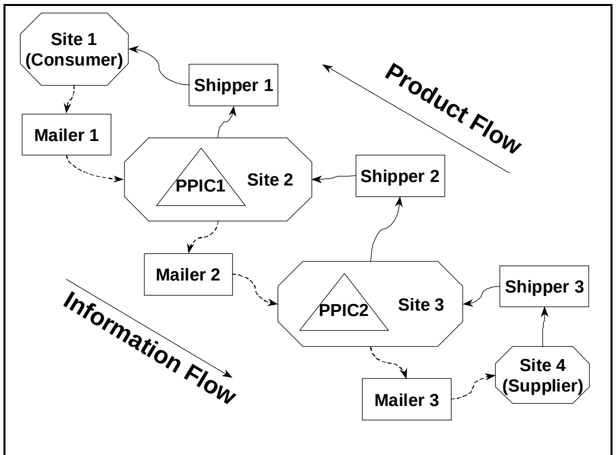

Figure 2.1: The DASCh Supply Chain

2.1.1. Consumer-1

Consumer-1 represents demand for the finished product. It sends orders to Site-2 and receives finished goods from Site-2. It normally expects orders to arrive as if shipped immediately from inventory of Site-2, and keeps statistics of average time to fill the orders and average order lateness.

2.1.2. Site-2

Site-2 represents the OEM's centralized shipping facility. It receives orders from Consumer-1 and fills them from its finished goods inventory as fast as it can. It orders goods from Site-3 using PPIC. The amount of time incoming goods must spend at a production site before they can be shipped out is a variable set by the user; in our experiments, we use a delay of 1. Site-2 may be subject to additional capacity constraints.

A separate object, PPIC-8, represents the forecasting and PPIC algorithms of Site-2.

2.1.3. Site-3

Site-3 represents the OEM's manufacturing facility. It receives orders from Site-2 and fills them from its finished goods inventory as fast as it can. It orders raw materials from Supplier-4 using PPIC. It takes a minimum of 2 time steps in our experiments to manufacture finished goods from raw materials. Site-3 may be subject to additional capacity constraints.

PPIC-9 represents the forecasting and PPIC algorithms of Site-3.

2.1.4. Supplier-4

Supplier-4 represents the supplier of raw materials. It receives orders from Site-3 and builds to order. In our experiments, we impose a delay of 4 time-steps from receipt of an order to shipment of finished goods. Supplier-4 is not subject to capacity constraints.

2.1.5. Order and Shipping Delays

It takes one time step for orders placed by Consumer-1, Site-2, or Site-3 to reach Site-2, Site-3, or Supplier-4, respectively. Similarly it takes three time steps (more generally, a mean and random variance) for goods to be shipped downstream from one entity to the next.

2.1.6. Order of Execution

Each time step the simulation does the following actions in order.

1. Orders and goods that are due to be delivered, are delivered.
2. Consumer-1 and the Sites run their PPIC algorithms and place their orders.
3. Sites add new finished goods to their inventories according to capacity constraints.
4. Sites and Supplier-4 send out shipments to fill orders that are due.

2.1.7. Initialization

The simulation is initialized to a steady state of orders and shipments of 100 units each time step, at each level in the supply chain. The Sites' PPIC algorithms behave as if they have seen a forecast of consumer demand for 100 units per time period up to and including time step 15. Thus in most situations one would set Consumer-1 to order exactly 100 units at least until time step 16. This cutoff is derived from certain delays in the system, and is specific to our experimental set-up, but can be changed as necessary.

2.2. Operation of the Simulation

2.2.1. The Configuration File

A configuration file is specified on the simulation command line. For example, the configuration file named "perfect.in" is specified via an argument "-IF=perfect.in". Much of the data in the configuration file is best left untouched, largely because the initialization routines make many assumptions about the configuration. Certain things noted below can only be changed in the configuration file.

2.2.2. Runtime Probes

In interactive mode, probes can be created to observe and modify parameters of the various entities. Some of the parameters are best left unchanged at runtime, however. For example, there are cases where arrays whose size depends on a parameter value, are allocated at the beginning of the simulation run and are not adjusted dynamically.

2.2.3. Consumer

The Consumer is capable of generating demand with noise, essentially rolling dice to yield the exact demand for each time step. It is also capable of sending an exact forecast of future orders to Site-2, for use when Site-2 is configured to receive a forecast from its customer. (For example, if the Consumer models an automotive OEM and Site-2 models a first tier supplier.) These facts imply that Consumer-1 must internally compute the actual amount it will order (rolling dice) ahead of time. The simulation generates Gaussian distributions specified by a mean and variance. If the random number generator returns a negative value, zero is used.

Table 2.1 summarizes the parameters in the Consumer agent.

The initial demand is specified in the *demandString* parameter in the configuration file. It is a string whose format is a series of triples separated by hyphens; each triple denotes a number of time steps, a mean, and a variance separated by colons. For example, “16:100:0-1:100:10”

Table 2.1: Consumer Parameters

Name	Type	Use	Meaning
demandString	String	Set in Configuration	Specifies demand. Triples separated by hyphens; each triple is number of time steps, mean, and variance.
cycleDemand	Unsigned	Set in both (see text)	Zero means the last mean and variance specified by demandString is repeated indefinitely. Nonzero means the demandString demand is repeated cyclically.
demandMean	Unsigned	Set in both (see text)	The mean used in calculating the demand when it doesn’t come from demandString.
demandVar	Unsigned	Set in both (see text)	The variance used in calculating the demand when it doesn’t come from demandString.
prevDemand	Unsigned	Read-only Probe	The actual demand of the previous time step.
nextDemand	Unsigned	Set in Probe	The amount that will be ordered by Consumer-1 next time step, regardless of what has been in the forecast sent out previously.
sourceLeadTime	Unsigned	Set in Configuration	If the Consumer sends an order at time step *t*, it expects to receive the goods at time *t + sourceLeadTime*. If set to 0, the sum of the means of the involved delays is used.
ppicUntil	Unsigned	Set in Configuration	The Consumer sends a forecast of its demand for *ppicUntil* time steps into the future, including the current time step. *ppicUntil* is enforced to be at least one. A value greater than one is useful only if the upstream Site’s PPIC uses Upstream Prediction forecasting. See the Perfect Prediction scenario, below.

means 16 steps of demand at 100 followed by one step of demand with a mean of 100 and variance of 10.

When the simulation starts up, the Consumer parses its *demandString* and stores the results in some internal arrays. Thus changing its value at runtime has no effect. Initially, the Consumer has to pre-compute its demand for the first *ppicUntil* time steps. In order to maintain a forecast *ppicUntil* steps long, each time step it has to compute the demand for the time *ppicUntil* steps in the future. Changing the values of *cycleDemand*, *demandMean*, and *demandVar* at runtime will not affect demands already computed, but will begin to affect the demand *ppicUntil* - 1 steps in the future. Note that if *cycleDemand* is one or if the means and variances specified in *demandString* have not yet run out, then changing *demandMean* and *demandVar* has no effect.

PrevDemand is available purely for informational purposes, to be viewed in a probe. It shows the actual demand used in the previous time step. Changing its value has no effect. *NextDemand* displays the actual demand that has been forecast and will be used in the next time step. Changing it actually changes the demand that is used, thus rendering the prior forecasts inaccurate.

The parameters *sourceLeadTime* and *ppicUntil* are analogous to like-named parameters in the Sites. *SourceLeadTime* is the number of time steps the Consumer expects it to take from when it sends out an order to when the goods are received. It is only used in computing lateness. If set to zero, the Consumer uses sum of the expected delays in transmitting the order to Site-1 and shipping the product back.

PpicUntil is the number of time steps of forecast demand that are made available to Site-1's PPIC algorithm. This determines the number of future time steps of actual demand that must be calculated in advance. See the discussion on the Upstream Prediction forecasting method and the Perfect Prediction scenario, below.

2.2.4. Site

Sites receive orders and shipments of their inputs at the beginning of each time step, run their PPIC algorithms, send out orders for their inputs, process WIP into finished goods inventory, and send out shipments of their products. A Site's goal is always to fill incoming orders from inventory, shipping on the same time step the order is received.

The model of processing at a Site is in two stages. First, incoming materials are "aged" or delayed to simulate the minimum amount of time it takes to do the processing. Second, the WIP is forced through a capacity constraint "funnel" that can only allow a maximum number of units to become finished goods each time step. Either stage may be circumvented: the "aging" can be set to zero and the capacity constraint can be set arbitrarily high. Gaussian noise may be added to either processing stage.

Table 2.2 shows the parameters for site agents.

Incoming materials are subject to "aging" in the *processingArea*. Then they are moved to *inProcess* where they are subject to the capacity constraint. When they make it through the

Table 2.2: Site Parameters

Name	Type	Use	Meaning
delayMean	Double	Set in Configuration	Mean number of time steps to "age" incoming materials to represent processing.
delayVar	Double	Set in Configuration	Variance in the Gaussian distribution of the number of time steps for "aging".
capacityMean	Double	Set in both	Mean number of units that can move from *inProcess* to finished goods *inventory* in one time step.
capacityVar	Double	Set in both	Variance in the Gaussian distribution of the number of units that can move to *inventory* in one time step.
capacity-WithNoise	Double	Read-only Probe	Actual capacity for the current time step (mean plus actual sampled variance)
capacity-MaxMultiplier	Double	Set in both	The capacity distribution is capped at *capacityMean * capacityMaxMultiplier*.
inProcess	Unsigned	Set in both	Total number of units that have been "aged" but not passed through the capacity "funnel".
inventory	Unsigned	Set in both	Finished goods inventory as a total number of units.
totalThroughput	Unsigned	Read-only Probe	Total actually processed up to current time step.

capacity "funnel", they are placed in finished goods *inventory* from where they are used to fill incoming orders from downstream.

A Site or Supplier does not ship materials for an order until it has sufficient finished goods inventory to fill the whole order. A Site sends orders for its materials to the upstream Site or Supplier, called its *source,* and receives a separate shipment for each order it sends. Each shipment of materials is treated as a single lot for the "aging" process. For example, if the Site's "aging" is set to a mean of 5 and a variance of 2, the distribution is sampled once for the whole lot, and the whole lot moves to *inProcess* for example 6 time steps after it is received.

Once the WIP reaches *inProcess*, it loses its identity as a lot and becomes merely a number of units. The capacity "funnel" allows a certain number of units, not lots, to move from *inProcess* to finished goods *inventory* each time step. Units from *inventory* are assembled into lots according to orders received from the downstream Consumer or Site. If a Site has multiple orders due (e.g. some are overdue), it will fill the oldest (by when received, not due date) for which it has sufficient inventory, first. It does not consider filling orders that, if shipped, would expect to arrive prior to the due date (using the expected shipping delay).

It is possible for a Site to receive a shipment of materials and, if the "aging" delay is zero and the capacity sufficiently high, in the same time step move it all the way through processing and ship out the resulting finished goods.

Each time a shipment of materials is received, the Site samples a Gaussian distribution with mean *delayMean* and variance *delayVar* to determine how many time steps to "age" the lot before adding it to *inProcess*. If the sample is less than zero, zero is used.

Each time step, the Site samples a Gaussian distribution with mean *capacityMean* and variance *capacityVar* to determine how many units to move from *inProcess* to *inventory*. If the sample is

Table 2.3: PPIC Forecast Parameters

Name	Type	Use	Meaning
forecastMethod	Unsigned	Set in both	0 = constant 1 = weighted average 2 = upstream prediction
forecastWindow	Unsigned	Set in both	The number of time steps of past actual demand used by the weighted average method.
forecastVar	Double	Set in both	Variance used when adding noise to the constant and upstream prediction methods.
historicDemand	Unsigned	Set in both	Mean demand of constant method and presumed demand prior to beginning of simulation run.
predictUntil	Unsigned	Read-only Probe	The number of time steps of forecast (including current time step) that are needed by the PPIC algorithm.

less than zero or greater than *capacityMean * capacityMaxMultiplier*, the distribution is resampled.

2.2.5. PPIC: Forecasting

Each Site delegates two of its functions to its corresponding PPIC (Production Planning and Inventory Control) agent. One is the development of a forecast of future incoming orders for its product and the other is running the PPIC algorithm to predict its future inventory and outgoing orders for materials.

Table 2.3 shows the parameters that the PPIC agents use in forecasting demand.

There are currently three methods for developing the forecast, determined by *forecastMethod*. If *forecastMethod* = 0, a constant value with Gaussian noise is used; if it is 1, a weighted average of past actual orders is used; and if it is 2, a prediction of future orders from the upstream Site or Consumer with Gaussian noise is used. The demand for the current time step is always known exactly: it is the sum of the due and overdue incoming orders already in hand. In some configurations it is possible to receive orders ahead of expectation and therefore have orders which are not due to be shipped immediately. For time steps after the current one, the maximum of the actual orders and the initial value determined by the forecasting method is used by the PPIC algorithm.

The forecast is made far enough into the future to enable the PPIC algorithm to compute the amount that will be ordered each time step, for *ppicUntil* time steps into the future (including the current time step). The order it sends out on that last time step will arrive after some delay as input materials, and after further delay will become finished goods ready to ship out. (The sum of those delays is called *myLeadTime*.) The forecast has to go out to the expected time step in which that last order will be available to ship as product; thus the forecast is for *ppicUntil* plus *myLeadTime* time steps.

2.2.5.1. Constant Method

The constant method samples a Gaussian distribution with mean *historicDemand* and variance *forecastVar*, once for each future time step, to determine the initial forecast. Although the sample could be negative, when it is compared with actual orders (which are nonnegative), the maximum will be nonnegative. The sampling is done again every time step; for example, at time $t = 10$, it will sample the distribution to predict demand for the future time $t = 15$. Later, at time $t = 11$, it will take a new sample of the distribution to predict demand for the future time $t = 15$.

2.2.5.2. Weighted Average Method

The weighted average method computes a weighted average of the actual demand for the past *forecastWindow* time steps and uses it as the initial forecast value for all future time steps. If t is the current time, $demand_{t-i}$ is the actual demand at time $t-i$, and w_i is the weight of the i^{th} previous demand, then the formula for the weighted average is

$$w_i = forecastWindow + 1 - i$$

$$\frac{\sum_{i=1}^{forecastWindow} w_i demand_{t-i}}{\sum_{i=1}^{forecastWindow} w_i} = \frac{2}{n(n+1)} \sum_{i=1}^{forecastWindow} w_i demand_{t-i}$$

When $t - i$ is before the simulation run began (i.e. $i > t$), *historicDemand* is used for the demand.

2.2.5.3. Upstream Prediction Method

When *forecastMethod* is 2, the forecast is based on a prediction passed up from the downstream Site's PPIC (or from the Consumer). The prediction gives the number of units that will be ordered each time step, starting with the current time and going on for the downstream entity's *ppicUntil* steps. For each future time step of the forecast, the downstream prediction is used as the mean of a Gaussian distribution and *forecastVar* as the variance; a sample is taken and used as the initial forecast. If the prediction does not extend far enough into the future, the last value is repeated. If the prediction is absent (only possible on the very first step of a simulation run), *historicDemand* is used.

Above it was mentioned that the forecast is developed for *myLeadTime* plus *ppicUntil* time steps. In order for the prediction passed up from the downstream PPIC or Consumer to be long enough to cover the entire forecast, the downstream entity's *ppicUntil* must be equal to or greater than *myLeadTime* plus *ppicUntil*. The Perfect Prediction scenario (below) gives an example of this.

In the current implementation, the prediction is attached to the actual order and carried with it. If the order delivery delay is increased and given noise, odd effects could result. The software is carefully written not to crash (famous last words), but the dynamics are likely to be illusory.

Table 3.4: PPIC Algorithm Parameters

Name	Type	Use	Meaning
safetyStock	Unsigned	Set in both	PPIC attempts to keep inventory at or above this level.
sourceLeadTime	Unsigned	Set in both	If the Site sends an order at time step *t*, it expects to receive the materials at time *t + sourceLeadTime*. If set to 0, the sum of the means of the involved delays is used.
sourceBatchSize	Unsigned	Set in both	Outgoing orders for materials are integral multiples of *sourceBatchSize.*
ppicUntil	Unsigned	Set in Configuration	The PPIC computes and sends a forecast of its demand for *ppicUntil* time steps into the future, including the current time step. It is enforced to be at least one. A value greater than one is useful only if the upstream Site's PPIC uses upstream prediction forecasting. See the Perfect Prediction scenario, below.
myLeadTime	Unsigned	Do not use!	If the Site will need to have more finished product at time step *t + myLeadTime*, it should order more materials at time *t*. If set to 0, the sum of the means of the involved delays (including *sourceLeadTime*) is used.

2.2.6. PPIC Algorithm

The PPIC algorithm is greatly simplified because there is a single product made from a single raw material obtained from a single source. Table 2.4 shows the parameters used in the PPIC algorithm. Starting at the current time step and going out into the future, it lays out

- the expected outgoing shipments (the forecast, which is the actual due and overdue orders for the current time step),
- the expected replenishments to inventory (product coming out of the corresponding Site's processing, ready to ship),
- the expected resulting inventory at the end of the time step, and
- the number of units of material that must be ordered each time step in order to keep the inventory at or above the *safetyStock* level.

The PPIC does not modify amounts or due dates on orders that have already gone out. It also does not take into account the capacity constraint. It does look at expected delays through the *sourceLeadTime* and *myLeadTime* parameters. When it orders, it only orders in integral multiples of *sourceBatchSize*.

The parameter *myLeadTime* should be replaced with a read-only output parameter that the user can view in a probe. Currently it should be set to zero in the configuration file, and only *sourceLeadTime* should be modified. Setting *myLeadTime* to a nonzero value will cause inconsistent behavior.

2.2.7. Supplier

The Supplier builds to order rather than to inventory. It receives orders, "ages" them to simulate processing, and ships materials with no capacity constraints and no "finished goods" inventory. The "aging" delay is determined by sampling a Gaussian distribution governed by two parameters, *delayMean* and *delayVar*. If the sample is less than zero, zero is used. Table 2.5 summarizes the parameters of the Supplier agent.

Table 2.5: Supplier Parameters

Name	Type	Use	Meaning
delayMean	Double	Set in Config.	Mean number of time steps to "age" incoming orders to represent processing.
delayVar	Double	Set in Config.	Variance in the Gaussian distribution of the number of time steps for "aging".
totalOrdersFilled	Unsigned	Read-only Probe	Total number of items actually shipped over time to date
cntOrdersFilled	Unsigned	Read-only Probe	Total number of orders fulfilled over time to date
currentShipment	Unsigned	Read-only Probe	Total number of items shipped at the current time step.

3. Theory

A general theoretical analysis of a complex system such as a supply network is not practical. However, it may be possible within certain restricted domains. When it is possible, it can help guide experimentation and provide insight into experimental results. This section provides two such analyses. The first is restricted to the behavior of the system within its linear domain, and provided predictions that we were able to confirm experimentally using the DASCh model. The second examines an extremely circumscribed class of behavior (inventory oscillations) in a non-linear domain of the model. In this case we observed the behavior initially in experiments, and then developed the theory to deepen our understanding of what was happening.

3.1. Linear Domain: Amplification and Correlation of Variance

3.1.1. Analysis

Consider our supply chain with all batch sizes set to 1, infinite (more or less) capacity, and initial conditions that support a steady state of customer orders of say, 100 units per time step. Now add Gaussian independent, identically-distributed (IID) noise to the customer demand. We are interested in the response of the system, as evidenced in the ordering patterns down the chain, to this IID noise. There are no other sources of noise or of uncertainty in the chain.

Let *L* be the lead time. Let *f* be the length of the historical epoch used to make forecasts. Consider an element, *k*, of the chain. (*k=1* is the customer.) *R(t)* is the order placed at time t by element *k* to its supplier (element *k+1*), which, assuming no uncertainty or noise in shipping or in other delay times will be delivered to that element's inventory at time *t+L*. In our simple PPIC the expression for *R(t)* can be written as

$$R(t) = \max\left[0, A - I(t) - \sum_{t'=1}^{L}(-F(t'+t) + P(t+t') + Q(t+t'))\right] \quad (1)$$

- *F(t)* is the forecast of what we expect will be ordered by our customer (element *k-1*).
- *P(t)* is what we know is in the pipeline which is under the control of element *k* (e.g., already being processing in the factory).
- *Q(t)* is what we expect will be delivered to inventory at time *t*, but is not under control of element *k*. That is, there could, in principle, be uncertainty about *Q(t)*, but not about *P(t)*. For the present purposes, the distinction between *P* and *Q* is irrelevant.
- *I(t)* is the inventory at time *t*.
- *A* is the safety stock level.

Now, for simplicity, first consider *k=2*. The forecast used in our simple PPIC is given by

$$F(t) = \max[C(t), O(t)]$$

where *O(t)* is the actual amount we *know* must be shipped from inventory (site *k*) at time *t*, and *C(t)* is a weighted average forecast based on *f* previous orders. In our simple PPIC *C(t)* is just

$$C(t) = \frac{2}{39 \cdot 40} \sum_{j=1}^{39} S(t-j) \times (40-j)$$

S(t) is the actual customer demand on day t, i.e., what is supposed to be shipped from inventory on day t). In our case, *S(t) = M +* $\eta(t)$, where *M* is the mean customer demand and $\eta(t)$ is IID (actual Gaussian in our case).

An important point here is that *F(t)* is the *same* for all *t* from now up to *L* steps in the future. Furthermore, if the size of the noise is not too large, then we should never have to invoke the max condition in the definition of *F(t)*: *O(t)* is driven primary by back-orders, but if we never run out of inventory, we shouldn't ever need to have *O(t)* greater than *C(t)*. With 30% or less variance, this doesn't seem to happen. So, for the purposes of this simple analysis, assume that *F(t) = C(t)*.

Now, with no uncertainty other than consumer demand, we can write (1) as

$$R(t) = A - I(t) - \sum_{t'=1}^{L-1} \Delta I(t+'t) + F(L) \tag{2}$$

Here *R(t) = M +* $\rho(t)$, and *F(t) = M +* $\delta(t)$, where *M* is the mean customer demand. If we plug in for *R* and *F* and eliminate the *M's*, we have

$$\rho(t) = A - I(t) + \sum_{t'=1}^{L-1} \left[\delta(t) - \rho(t+t'-L)\right] + \delta(t) \tag{3}$$

Note that this is a linear auto-regressive expression for $\rho(t)$. Unlike $\eta(t)$, the driving term $\delta(t)$ is *not* IID. Rather it is a linear combination of IID terms.

I(t), the inventory at time *t*, is just

$$I(t) = I(0) + \sum_{t'=1}^{t-1} \Delta I(t') = I(0) + \sum_{t'=1}^{t-1} - S(t') + R(t'-L) = I(0) + \sum_{t'=1}^{t-1} \eta(t') - \rho(t'-L)$$

(Of course, we have to cut this off at zero, or let the lower limit go to -∞.)

Plug this into (3) and we have an expression for the sum over $\rho(t)$:

$$B(t) \equiv \sum_{t'=0}^{t} \rho(t') \ = L\delta(t) + A - I(0) + \sum_{t'=1}^{t-1} \eta(t')$$

B(t)-B(t-1) = $\rho(t)$, so we have:

$$\rho(t) = L(\delta(t) - \delta(t-1)) + \eta(t-1)$$

To compute the variance of $\rho(t)$ and imagine that we are doing an average over an ensemble, yielding

$$\langle \delta^2(t) \rangle \sim L^2 Q \langle \eta^2 \rangle$$

where Q is a numerical factor, discussed further below.

3.1.2. Discussion

Note three important features:

1. The variance of $\rho(t)$ is magnified by the factor of L^2 over the variance of η, because the algorithm uses the same $F(t)$ for all dates between now and L days in the future. The amplification points up the importance of maintaining appropriate characteristics in the forecasting function.

2. The variance is stationary.

3. The $k = 2$ order fluctuations about M, $\rho(t)$, are linearly correlated. Recall that $\eta(t)$ is IID and so not correlated, but the $k = 2$ order fluctuations are. In general, with a forecast that depends linearly on f terms in the past, $\rho(t)$ will be linearly correlated over a range of order f. This is important. The order fluctuations down the chain are not independent, even if the driving customer orders are.

It is straightforward to apply this analysis to other elements down the chain. The only thing that changes, semi-quantitatively, is that the driving orders to a $k>2$ element are not IID. Looking back at (2) one can deduce the following general characteristics:

- Let $L(i)$ be the ordering delay at site i and $f(i)$ be the historical horizon for forecasting at site i. Then, the variance of the fluctuations about the mean of orders placed by site k, is of order

$$\langle \rho_k^2(t) \rangle \sim L_1^2 \cdot L_2^2 \cdot L_k^2 \; Q' \langle \eta^2 \rangle \sim L_k^2 \; \overline{Q} \langle \rho_{k-1}^2(t) \rangle$$

  $\overline{Q}$ and Q' are numerical factors, discussed below.

- These fluctuations are linearly correlated over a time horizon of the order of

$$\sum_{i=1}^{k} f(i)$$

The inventory at time t, $I(t)$, shares some characteristics of the ordering time series. For example, at site $k = 2$, and under our assumptions above, $I(t)$ is given by

$$I(t) = I(0) + \sum_{t'=1}^{t-1} \Delta I(t') = \sum_{t'=1}^{t-1} \rho(t'-L) - \eta(t') = A + L\delta(t - L - 1)$$

The fluctuations of *I(t)* about *A* are also linearly correlated over times of order *f*. The variance of these fluctuations is stationary, and the magnitude of the variance also grows as we go down the chain, again like L^2. For sites with $k>2$, the range of linear correlation as well as the times over which these variances are linearly correlated grows in the same way (semi-qualitatively) as the orders placed time series.

Additional Comments:

1. In addition to the variances growing, the L^2, the fluctuations about the mean are linearly correlated. The existence of these dynamical correlations is not widely appreciated in industry, but they are clearly very important in analyses and ordering policy decisions.

2. As we move down the chain, the situation becomes more complicated, because the driving force for sites $k>2$ are already linearly correlated. Since all the correlations are linear, and since the equations are additive, it is unlikely that further analysis will derive any nonlinearities here. But if we want to do a detailed analysis of the magnification factors, we do need to consider carefully the correlations in the driving force.

3. The increase in the variance each time we move down the chain increases by a factor of L^2 and another numerical factor that we have called *Q*. *Q* depends in detail on the forecasting formula, and on whether the ordering data coming into site *k* are correlated or not. In our case, for example, for site 2, this factor is fairly small, and accounts for the fact that although the variance is larger for site 2 than for the customer orders, it is not 25 times larger. To compute this factor, just take the formula for $\delta(t)$, write it in terms of $\eta(t)$, form $\delta(t) - \delta(t-1)$, square it, and do an ensemble average remembering that $\eta(t)$ is IID. All the cross terms will vanish, and you will be left with a fairly small number. When we compute the variance of the orders placed by site 3, one difference is that the orders coming from site 2 are linearly correlated, and so the cross terms in the analogous calculation for site 3 do not vanish. Thus, the factor is not as small (although still less than one). Similar comments apply to the calculation of the variance of the fluctuations of the inventory. Here, though the factors are not as small since we need to square $\delta(t)$, not $\delta(t) - \delta(t-1)$.

4. Equations (1) and (2) can be considered particularly simple examples of feedback. That is, what you order on day *j* affects what you will order on day $j+i$, for some set of *i*. The feedback in this case is simple and linear. But it can easily become more interesting and nonlinear once we take capacity constraints, batching, and noise in other parts of the system into account.

5. The basic message here is that this system should be viewed as a dynamical system, in general, a nonlinear dynamical system, capable of a wide variety of behaviors. We have just seen the absolutely simplest one here, but even in this case, the analysis has important lessons for any company using a PPIC algorithm similar to the simple one we have used in this model.

6. From a research point of view, we have here a system with a range of parameters that can be adjusted to explore different regions. The region discussed here is a simple linear one. It is, in fact, with some effort, completely analytical. When we invoke capacity

constraints, other kinds of forecasting formulae, batching effects, as well as other kinds of driving forces, we will get into nonlinear regimes. There we will be able to do other kinds of analyses, but they will not be completely analytical as the linear one here. Potentially even more interesting is the fact that using this same basis, we can model the effects of various kinds of adaptive decision making.

3.2. Nonlinear Domain: Inventory Oscillations

Consider a supply chain in which consumer demand is constant and supplier capacity infinite, but in which the intermediate sites have capacity limitations lower than the level of consumer demand. In such a configuration,

- production sites will operate at capacity;
- finished goods inventory at production sites will build up until it is high enough to satisfy an order;
- at that point an order will ship and inventory will drop, then build up again until another shipment is possible.

Thus we expect to see inventory oscillations at sites with insufficient capacity, and our experiments bear this out. This section establishes a representation for such oscillations and definitions, then makes a series of predictions that are satisfied by our experiments. (In fact, historically, the experiments came first, and observation of the regularities we observed in them led to this analysis.) Finally, it offers a useful geometric interpretation.

In this mode of operation, a useful abstraction of the model is the modulo function. Since each time step generates new inventory of *capacity* and outstanding orders ship everything in excess of *order*, the inventory at the *n*th time step is just *mod((n-1)*capacity, order)*, where *mod()* is the modulo function, the essence of a threshold nonlinearity. (Later, we will point out some details of the behavior of the system in this regime that are more complex than this simple abstraction.)

3.2.1. Representation and Definitions

A useful abstraction of the behavior of a given system consists of a numerical sequence describing the number of time steps needed to reach successive local maxima until the system returns to a previous level of inventory. That is, for a given *Demand* and *Production*,

1. Pick a local maximum in the inventory time series.
2. Record the number of steps needed to reach the next local maximum.
3. Repeat until the inventory is the same that it was at the original maximum.

For example, consider a system with *Demand* = Initial inventory = 170 and *Production* = 100. Experiment shows that successive inventory levels will be (170, **100**, *30*, **130**, *60*, **160**, 90, *20*, **120**, *50*, **150**, 80, *10*, **110**, *40*, **140**, 70, *0*, **100**). The local maxima are indicated by **bold-faced** numbers, and represent the points at which inventory rises above demand so that a shipment can take place. The sequence of steps-to-next-local-maximum, beginning with the first local maximum at 200 and continuing until inventory returns to 200, is (2,2,3,2,3,2,3). It is provably the case (shown below) that the same sequence will be generated if instead of focusing on local maxima, one focuses on local minima (the *italicized* numbers in the example series).

For some purposes, it is important to remove common factors from the ratio *Demand/Production*. Also, for the purpose of this analysis, we consider only one manufacturing site.

Definitions:

Demand: The (constant) level of orders from the consumer (site 1).

Production: The (constant) capacity level at the producer (site 2). By hypothesis, *Production < Demand*.

Inventory(t): The finished goods inventory at the producer (site 2). Where there is no danger of confusion, the temporal argument may be omitted.

D: The numerator of *Demand/Production* with all common factors removed (in the example, 17). Note that *D* is a legitimate *Demand*, but an arbitrary *Demand* may not be a legitimate *D*.

P: The denominator of *Demand/Production* with all common factors removed (in the example, 10). Note that *P* is a legitimate *Production,* but an arbitrary *Production* may not be a legitimate *P*.

H: The minimum of *P* and $D - P$. In the case that these are equal, observe that $H = D/2$ (which motivates the abbreviation *H[alf])*. Since by construction there are no common factors in *D* and *P*, $H = D/2 \rightarrow D = 2 \; \& \; P = 1$.

I(t): The finished goods inventory at the producer, scaled by any factors removed from *Demand* and *Production*: *Inventory(t)*D/Demand = Inventory(t)*P/Production.* Where there is no danger of confusion, the temporal argument may be omitted.

The next three definitions presume that the system has entered the region $P \leq I < (D+P)$. We demonstrate in the next section that it will enter this region, and that once there, it will remain there.

Sequence: The shortest sequence of steps-to-next-local-maximum between two equal inventory levels at the producer; in the example, (2, 2, 3, 2, 3, 2, 3).

Period: The minimum number of time steps such that $I(t) = I(t+Period)$ (the sum of the elements of *Sequence*; in the example, 17).

Length: The number of elements in *Sequence* (in the example, 7).

Ceil(n): The least integer greater than or equal to n.

Floor(n): The greatest integer less than or equal to n.

3.2.2. Predicted Behaviors

We predict the following behaviors, all of which are observed experimentally.

3.2.2.1. Attractor

If the system is initiated with $Inventory \geq Demand$, it will enter the region $0 \leq Inventory < Demand$. Once the system enters this region, it will remain there.

Proof: First we show that the system will enter the attracting region if it is started outside. Assume that the system is initiated with *Inventory(0)* $\geq$ *Demand.* Because *Demand* > *Production*, inventory will drop by *(Demand – Production)* at each time *t* for which *Inventory(t)* $\geq$ *Demand*, until *Inventory* < *Demand*.

Now we show that the system remains in the attractor once there. Consider two cases.

1. *Inventory(t)* < *Demand* - *Production*. Then no shipment can be made, and *Inventory(t+1) = (Inventory(t) + Production) < Demand.*
2. *(Demand – Production)* $\leq$ *Inventory(t)* < *Demand*. Then some of the inventory is used to make up the defect in production, so *Inventory(t+1) = (Inventory(t) – (Demand - Production)).* Since *Production* < *Demand*, this level will be strictly less than *Demand.* Furthermore, even if *Inventory(t) = (Demand – Production)* (the lowest it can be in this case), *Inventory(t+1) = Demand - Production – Demand + Production = 0,* and if *Inventory(t) > Demand,* then *Inventory(t+1) > 0.*

Thus in both cases the system remains in the region *0* $\leq$ *Inventory* < *Demand*. □

3.2.2.2. Scaling

If we multiply *Demand* and *Production* by the same integer factor, or if we divide out common integer factors, the series *Inventory(t)* (and thus the attracting region *0* $\leq$ *Inventory* < *Demand*) is multiplied or divided by the same integer factor, but *Sequence* and *Period* are unaffected.

Proof: The proof rests on a manufacturing interpretation of what means to multiply or divide *Demand* and *Product* by a common factor *k*. If there exists such a common factor, division means that it is possible for the producer to package the individual products in bundles of *k*, and for these bundles to be delivered intact to the consumer, without changing the total amount produced or shipped in each unit of time. Multiplication by an arbitrary common factor *k* means that each product is in fact an assembly of *k* parts, and we now agree to count the parts individually rather than as assemblies. In neither case do we actually change the amount of product manufactured, or the time it takes to manufacture it. However, we do change the units in which we count the production. Thus measures in units of time (including *Period* and the steps between local maxima in *Sequence*) are not affected, but measures in units of product (*Demand, Production,* and *Inventory(t)*) will be *k* times smaller (for division) or larger (for multiplication) than previously. □

Note: This principle motivates the use of *D* and *P*, from which all common factors have been removed, as a unique representation of a given ratio *Demand/Production*.

3.2.2.3. Period

For any *I(t)* in the region $0 \leq I < D$, the system will return to the same inventory level at time $t+D$, so that *Period = D.*

Proof: This proof rests on the lack of common factors in the representation *D/P* of the demand/production ratio. We first show that *Period* $\leq$ *D,* then eliminate the inequality.

1. Assume $I(t)$ is in the region. After D steps, the producer will have produced $D*P$, which is exactly divisible by D and so will all have been shipped, leaving $I(t + D) = I(t)$. So *Period* cannot be greater than D.
2. Assume $Period < D$ and derive a contradiction. Since $I(t) = I(t+Period)$ by the definition of *Period*, the amount manufactured during *Period*, which is $Period*P$, must be divisible by D. However, neither *Period* nor P is individually divisible by D (both being by hypothesis less than D). This means that *Period* and P have factors a and b, respectively, such that $D = a*b$. But then both D and P are divisible by b, contrary to our assumption that all common factors have been removed. So the assumption $Period < D$ must be wrong.

Having ruled out both $Period > D$ and $Period < D$, we conclude $Period = D$. □

Note: By *Scaling* (Section 3.2.2.2), $Period = D$ not only for systems in the (D,P,I) units, but for arbitrarily scaled *(Demand, Production, Inventory)* units. For example, *Period* for the system presented above with *Demand* = 170 and *Production* = 100 is 17, the same as for $D = 17$ and $P = 10$.

3.2.2.4. Coverage

Between t and $t + Period$, I assumes every value in the range $0 \leq I < D$.

Proof: There are D values in the range $0 \leq I < D$, and $Period = D$ time steps must pass before any repeats, therefore at each time step *Inventory* must assume a different value, and all D values will be required to consume all D time steps. □

Note: This result holds only for the reduced units (D, P, I), since it concerns units of parts produced. For systems in which *Demand* and *Production* have a common factor k, there will be bands of inventory values of width k that the system will never visit once it is in the attracting region.

3.2.2.5. Relation of Local Minima and Maxima

The pattern by which $I(t)$ moves between local minima and local maxima in the attracting region depends on H. There are three cases.

1. If $H = P = D\text{-}P = D/2$, $I(t)$ is always either at a local minimum or a local maximum. If $I(t)$ is a local minimum, then $I(t+1)$ is a local maximum. If $I(t)$ is a local maximum, then $I(t+1)$ is a local minimum.
2. If $H = P < D\text{-}P$, then if $I(t)$ is a local maximum, $I(t+1)$ is a local minimum; if $I(t)$ is a local minimum, then $I(t+p)$ is a local maximum, where p is either $Ceil(D/H) - 1$ or $Ceil(D/H) - 2$. If $H = 1$, $p = D/H – 1$ uniquely.
3. If $H = D\text{-}P < P$, then if $I(t)$ is a local minimum, $I(t+1)$ is a local maximum; if $I(t)$ is a local maximum, then $I(t+p)$ is a local minimum, where p is either $Ceil(D/H) - 1$ or $Ceil(D/H) - 2$. If $H = 1$, $p = D/H – 1$ uniquely.

Proof: Consider the three cases separately.

1. Assume $H = P = D\text{-}P = D/2$. Notice in particular that $P = D\text{-}P$. Consider first the case where $I(t)$ is a local minimum. Thus the net change in the next time step must be positive, which means there can be no shipment. Therefore $I(t+1) = I(t) + P \geq D\text{-}P$, which will permit a

shipment, and thus be a local maximum. Now consider the case where $I(t)$ is a local maximum, thus $D\text{-}P \leq I(t) < D$, and a shipment will take place. The net change in inventory will be a reduction of $(D\text{-}P) = D/2$. As a result, $I(t+1) < D/2 = D\text{-}P$, so no shipment will be possible, and $I(t+1)$ is a local minimum. Thus local minima and maxima alternate at each time step.

2. Assume $H = P < D\text{-}P$. Consider first the case where $I(t)$ is a local maximum. Therefore $D\text{-}P \leq I(t) < D$, and the net change in inventory with the next shipment will be a reduction of $D\text{-}P$, leaving $0 \leq I(t+1) < D\text{-}(D\text{-}P) = P < (D\text{-}P)$. No shipment is possible, so the system is at a local minimum. Now let $I(t)$ be at such a local minimum, $0 \leq I(t) < P$. The next local maximum will occur when $D\text{-}P \leq I(t+p)$. The net change in inventory needed to satisfy this condition $\Delta I = I(t+p) - I(t)$ will fall in the range $(D\text{-}2P) < \Delta I \leq (D\text{-}P)$. The number of time steps needed to make this shift, at P units per time step, is thus in the range $Floor((D\text{-}2P)/P) = Floor((D\text{-}2H)/H) = Floor(D/H)\text{-}2 < p \leq Ceil((D\text{-}P)/P) = Ceil((D\text{-}H)/H) = Ceil(D/H)\text{-}1$. $D/H = D/P$ can be integral only in the case that $P = 1$, in which case $D/H\text{-}2 < p \leq D/H\text{-}1$, leaving $p = D/H\text{-}1$. Otherwise, $Floor(D/H) = Ceil(D/H)\text{-}1$, so $Ceil(D/H)\text{-}3 < p \leq Ceil(D/H)\text{-}1$. Thus $p = Ceil(D/H) – 1$ and $p = Ceil(D/H) - 2$ are the only two options.

3. Assume $H = D\text{-}P < P$. Consider first the case where $I(t)$ is a local minimum. Therefore $0 \leq I(t) < D\text{-}P$, and the net change in inventory with the next shipment will be an increase of P, leaving $P \leq I(t+1) < D$. Since $P > (D\text{-}P)$, a shipment is possible, and the system is at a local maximum. Now let $I(t)$ be at such a local maximum, $P = D\text{-}H \leq I(t) < D$. The next local minimum will occur when $I(t+p) < D\text{-}P = H$. The net change in inventory needed to satisfy this condition $\Delta I = I(t) - I(t+p)$ will fall in the range $(D\text{-}2H) < \Delta I \leq (D\text{-}H)$. The number of time steps needed to make this shift, at $D\text{-}P = H$ units per time step, is thus in the range $Floor((D\text{-}2H)/H) = Floor(P/H)\text{-}2 < p \leq Ceil((D\text{-}H)/H) = Ceil(D/H) - 1$. If D/H is non-integer, $Floor(D/H)\text{-}2 = Ceil(D/H)\text{-}3 < p \leq Ceil(D/H)\text{-}1$, leaving $p = Ceil(D/H) – 1$ and $p = Ceil(D/H) - 2$ as the only two options. But the only way $D/H = D/(D\text{-}P)$ can be integral is if $H = 1$. To see this, assume that $D/(D\text{-}P) = n$. If $n = 2$, we have $P = D\text{-}P$, which we have already considered in case 1. So $n > 2$. On our assumption, $D = n(D\text{-}P)$, or $nP = (n\text{-}1)D$, which can only be true if either $n/(n\text{-}1)$ is integral (which it is not for $n > 2$), or there exist factorizations $P = a(n\text{-}1)$ and $D = bn$ for integral a,b. Substituting these into $nP = (n\text{-}1)D$, we have $na(n\text{-}1) = (n\text{-}1)bn$, or $a = b$, which would mean that D and P have a common factor $a = b$. If this factor is greater than 1, we have a contradiction with our assumption that D and P have no common factors. So $a = b = 1$. Then $D = n$, $P = n\text{-}1$, and $H = D\text{-}P = 1$. In this case, $p = D/H\text{-}1$ uniquely

These three cases exhaust the possibilities. □

Note 1: In case 1, the removal of common factors means that $D = 2$ and $P = H = 1$.

Note 2: In the last two cases, p is the number of steps from a local extremum of one kind to a neighboring extremum of the opposite kind. Each entry in *Sequence* is from local maximum to local maximum, and thus equal to $p+1$, restricting it to be either $Ceil(D/H) – 1$ or $Ceil(D/H)$. If $H = 1$, then all entries in *Sequence* are D.

Note 3: In each case, local minima are adjacent to local maxima, and always (for a given case) in the same direction. Thus one could measure time steps between local minima instead of between local maxima, without changing the results, and such a procedure would yield the same *Sequence*.

3.2.2.6. *Length*

Length, the number of items in the sequence, corresponding to the number of intermediate maxima between maxima of the same size (counting one of the ends), is *H*.

Proof: The total production during a sequence is $Period*P = D*P$. To return to the same inventory level, the sum of shipped orders must be the same. Consider the same three cases analyzed in *Relation*.

1. Assume $H = P = D\text{-}P = D/2$. There is one shipment of size *D* every maximum, so total shipments are $D*Length$, which must equal production, $D*P$, yielding $Length = P$. But in this case $P = H$, so $Length = H$.
2. Assume $H = P < D\text{-}P$. Again, there is one shipment for every maximum, so $D*Length = D*P$ and $Length = P$, and again $P = H$, so $Length = H$.
3. Assume $H = D\text{-}P < P$. Now there is only one step during each maximum when there is not a shipment, so the total number of shipments is $(D\text{-}Length)$ and the total amount shipped is $(D\text{-}Length)*D$, which must equal production, $D*P$. Thus $(D\text{-}Length) = P$, $Length = (D\text{-}P)$, but in this case $(D\text{-}P) = H$, so $Length = H$.

These three cases exhaust the possibilities. □

3.2.2.7. *Proportion of Long and Short Periods*

When $H \neq 1$, the periods of the *H* extrema of the same kind (the *H* maxima or the *H* minima) in a sequence are not all equal. $H*Ceil(D/H) – D$ have period $Ceil(D/H) – 1$ and $D – H*Floor(D/H)$ have period $Ceil(D/H)$.

Proof: $Period = D$ is equal to the sum of the elements of *Sequence*, which (by *Note 2* to Section 3.2.2.5) can only be of two periods, $Ceil(D/H) – 1$ or $Ceil(D/H)$. Note first that both kinds of periods must appear, for when $H \neq 1$, $Ceil(D/H) > D/H$ so that $H*Ceil(D/H) > D = Period$, while $Ceil(D/H)\text{-}1 < D/H$ so that $H*(Ceil(D/H)\text{-}1) < D = Period$.

To find the proportion of periods of each length, let *A* be the proportion of maxima of length $Ceil(D/H) – 1$. Then $H\text{-}A$ are of length $Ceil(D/H)$, and $Period = D = (H\text{-}A)*Ceil(D/H) + A*(Ceil(D/H) – 1) = H*Ceil(D/H) – A$, or $A = H*Ceil(D/H) – D$ and $(H\text{-}A) = D – H*(Ceil(D/H) – 1) = D – H*Floor(D/H)$ (where the conversion from *Ceil* to *Floor* uses the assumption that $H \neq 1$). □

3.2.2.8. *Monotonic Subsequences*

In the case that $H \neq 1$, the number of monotonic subsequences in the overall *Sequence* is equal to the lesser of $H*Ceil(D/H) – D$ and $D – H*Floor(D/H)$ (that is, the number of extrema with the less common period).

Proof: It suffices to show that no two extrema with the less common period are adjacent. Given this result, the less common extrema must be distributed among the more common ones, generating that many monotonic subsequences in *Sequence*. Recall from *Relation* that the cycles

have two shapes. If $H = P < D-P$, the movement from maximum to maximum consists first of a drop of $D-P$ followed by p climbs of P (the "ascending" configuration), where p is either $pShort = Ceil(D/H)-2$ or $pLong = Ceil(D/H)-1$, for a net gain over a cycle of $(p+1)*P - D$. If $H = D-P < P$, the movement from maximum to maximum consists first of p drops of $(D-P)$, where p is either $pShort$ or $pLong$, followed by a climb of P (the "descending" configuration), for a net gain over a cycle of $(p+1)*P - pD$. These same differences also obtain if we count between successive minima. Our basic proof schema is to begin at a maximum $(D-P) \le I(t) < D$ or a minimum $0 \le I(t) < (D-P)$. We have to show that no two minority periods can fall together, so we assume that they do, and derive a contradiction to known properties of the system (such as showing that the inventory would leave the *Attractor* region $P \le I < (D+P)$, or that an extremum would occur outside of the appropriate region). Consider four cases, generated by pairing the option expressing which period of maximum is less common
short: $H*Ceil(D/H) - D < D - H*Floor(D/H)$
long: $H*Ceil(D/H) - D > D - H*Floor(D/H)$
with the option describing the pattern:
ascending: $H = P$
descending: $H = D-P$.

1. *Short, ascending* (example: $D/H = 17/6$.) (The proof formalizes the observation that in the short ascending case, each ascending cycle leads to a maximum that is lower than the previous maximum.) Observe that Two successive short ascending cycles from a maximum would generate a change in I, $\Delta I = 2[(pShort+1)*P - D] = 2[Ceil(D/H)-2+1)*P-D] = 2[P*Ceil(D/H)-P-D]$. Since any maximum is strictly less than D, after this change, I satisfies:

$I < D+2[P*Ceil(D/H)-P-D]$	
$= H*Ceil(D/H) + (H*Ceil(D/H) - D) - 2P$	(using *ascending*)
$< H*Ceil(D/H) + D - H*Floor(D/H) - P - H$	(using *short* and *ascending*)
$= D-P$	(since, for D/H not integer, $Ceil(D/H)-Floor(D/H) = 1$)

   This last result asserts that after two short ascending cycles from a maximum, $I<D-P$. But then no shipment can take place, so we are not at a maximum, and the second period is not yet complete, contradicting our assumption.

2. *Short*, *descending* (example: $D/H = 17/11$.) (The proof formalizes the observation that in the short descending case, each descending cycle leads to a maximum that is higher than the previous maximum.) Two successive short descending cycles from a minimum give a change in I $\Delta I = 2[P*(Ceil(D/H)-1) - D*(Ceil(D/H)-2)]$. Since any minimum is greater than or equal to 0, the inventory I at the second minimum satisfies:

$I \ge 2(-H*Ceil(D/H) + H - D)$	(using *descending* in the form $P = D-H$)
$= 2(-H*Floor(D/H) - D)$	(using $Ceil(D/H)-1 = Floor(D/H)$ for non-integer D/H),
$> 2(H*Ceil(D/H) - 2D)$	(using *short*)
$\ge 2(3D - 2D) = 2D$	(recognizing that for $H < D/2$, $Ceil(D/H) \ge 3$)

But this last result asserts that after two short descending cycles from a minimum, $I > 2D > D+P$, contrary to *Attractor*, furnishing the desired contradiction.

3. *Long, ascending* (example: $D/H = 17/5$.) (In the long ascending case, each ascending cycle leads to a maximum that is higher than the previous maximum.) Two successive long ascending cycles from a maximum generate a change in I, $\Delta I = 2[(pLong+1)*P - D] = 2[Ceil(D/H)-1+1)*P-D] = 2[P*Ceil(D/H)-D]$. Since any maximum is greater than or equal to $D-P$, after this change, I satisfies:

$I \geq D-P+2[H*Ceil(D/H)-D]$	(using *ascending*),
$> H*Ceil(D/H) + D - H*Floor(D/H) - P$	(using *long*),
$= D$	(since $Ceil(D/H)-Floor(D/H) = 1$ for D/H not an integer, and again using *ascending*).

But this last result asserts that after two long descending cycles from a maximum, $I > D$, contrary to *Attractor*, furnishing the desired contradiction.

4. *Long, descending* (example: $D/H = 17/10$.) (In the long descending case, each descending cycle leads to a maximum that is lower than the previous maximum.) Two successive long descending cycles from a minimum give a change in I $\Delta I = 2[P*(Ceil(D/H)) - D*(Ceil(D/H)-1)]$. Since any minimum is strictly less than $D-P$, the inventory at the second minimum satisfies:

$I < D-P + 2(-H*Ceil(D/H) + D)$	(using *descending*)
$= 3D - H*Ceil(D/H) - H*Ceil(D/H) - P$	
$= 3D - H*Ceil(D/H) - H*Floor(D/H) - H - P$	(using $Floor(D/H) = Ceil(D/H)+1$ for D/H not an integer)
$< 3D - H*Ceil(D/H) + H*Ceil(D/H) - 2D - H - P$	(using *long*)
$= D-H-P = 0$	(using *descending*).

But this last result asserts that after two long descending cycles from a minimum, $I < 0$, contrary to *Attractor*, furnishing the desired contradiction.

These four cases exhaust the possibilities, establishing the desired result. ☐

3.2.3. A Geometrical Interpretation

The behavior outlined in the previous section is consistent with a concise geometrical model of the dynamics.

The behaviors demonstrated above show that the complete dynamics can be represented in a square of D units on a side. The left edge of the square corresponds to time t, the right edge to time $t+D$, the bottom to inventory P, and the top to inventory $D+P$. Let t be the time at which inventory first falls within the *Attractor*. At each time step draw a line segment of slope P and length $\sqrt{P+1}$ beginning at $I(t)$ to define $I(t+1)$.

As long as $I < D$, the end of this line segment will fall within the square, and will correspond to the trajectory of the inventory. If $I(t) \geq D$, a distance $d_1 = D+P-I(t) < P$ from the top edge of the square, the end of such a line segment would fall at or beyond that edge, outside the range of the *Attractor*. With the real system, in this case the new inventory is $I(t)-D+P$, which is $d_2 = I(t)-D$

units above the bottom edge of the square. Note that $d_1+d_2 = P$, which means that if we connect the top and bottom edges of the square to form a cylinder, then beginning at any valid *I(t)*, the next *I(t)* can be found by appending such a line segment (in effect, wrapping a string around the cylinder at a constant angle.

From the *Period* behavior (Section 3.2.2.3), we know that $I(t) = I(t+D)$, and the determinism of the system means that the trajectory then repeats. In terms of our geometric model, this behavior corresponds to connecting the ends of the cylinder to each other to form a torus.

In our manufacturing domain, *D* and *P* are integer parameters, so *D/P* is rational by construction. However, the torus model supports irrational *D/P* as well. In this case, we would have quasiperiodicity, and the orbit on the torus would never retrace itself.

4. Experimental Results

Experiments with the system described in the previous Part 2 show four effects involving variation within the supply chain, three of which have not been discussed in any detail in the previous literature. The first three of these effects are observed within the linear domain of the model, and so are in principle susceptible to analytic treatment. The fourth results from imposing nonlinear constraints on the system. As these results emerged, we reviewed them regularly with our informant from the Fortune-100 electronic manufacturer. He recognized them as characteristic of real manufacturing systems, but credited the model with making him aware that they could be generated by such simple mechanisms.

4.1. Amplification of Variation

Variation amplification has been widely discussed in the literature, as discussed in “Previous Work: Observed Behavior” above. It is well recognized, emerges from our theoretical analysis (Section 3.1), and was an expected result. Our observation of it helps to validate our approach. In addition, we observe different dynamics in the upper (distribution) and lower (input) halves of the supply chain hourglass.

The symptom is that subtier suppliers see more variability in the orders they get than the OEM generates in its orders to the first site in the supply chain. To experiment with this dynamic we set up a configuration where all the batch sizes are one, so the economic order quantity does not introduce a nonlinearity. The orders are generated by the top-level customer at a rate of 100 per week with a IID (Independent, Identically Distributed) variance of 10 per week. The capacity is set at 10,000 per week, virtually infinite in comparison with the order levels, again avoiding a threshhold nonlinearity.

4.1.1. Distribution Networks (Upper Half of Hourglass)

Using the weighted forecasting method appropriate for the upper half of the supply chain hourglass, Figure 4.1 shows the mean and variance of the weekly orders in an experimental run of 500 weeks from each of the top three sites in the model. The mean is constant at 100, but the variance grows dramatically from 10 in the orders issuing from the consumer to 14.5 in those that Site 2 sends, and then to 30.0 in those from Site 3.

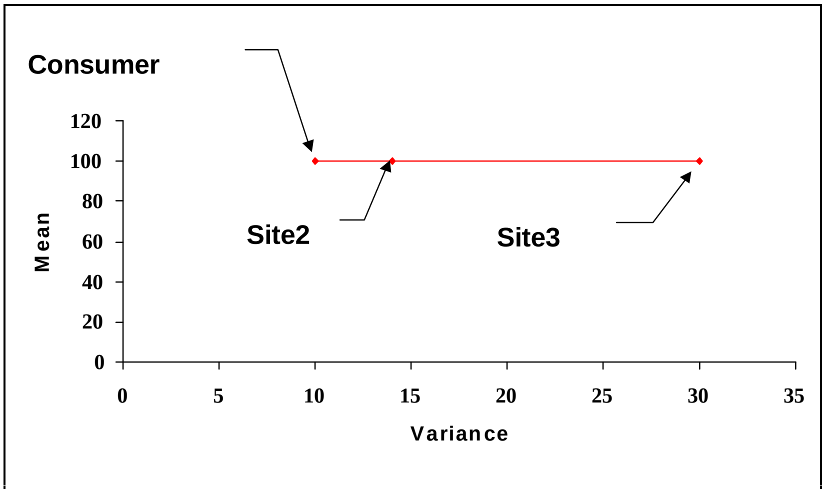

Figure 4.1: Variance Amplification

Our theoretical analysis (Part 3) predicts that variance in the orders coming from intermediate sites should be proportional to the variance in their incoming order streams, times the square of the lead times that they see. To test this analysis, we generated a series of emulation runs with

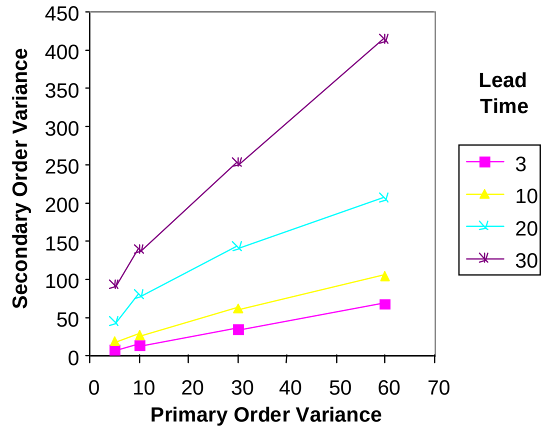

Figure 4.2: Dependence of Site 2 Variance on Consumer Variance

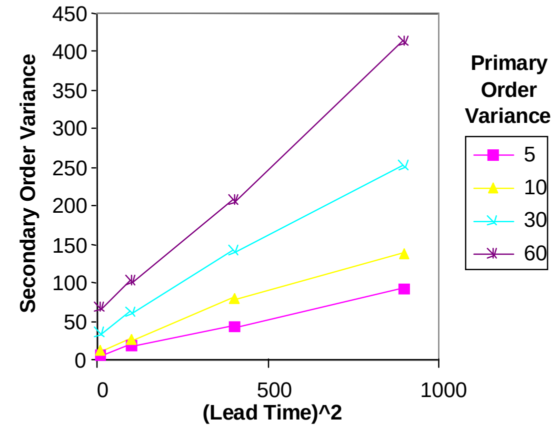

Figure4.3: Dependence of Site 2 Order Variance on Lead Time

consumer variances of 5, 10, 30, and 60 and lead times of 3, 10, 20, and 30 time steps. We also explored varying the forecast window in the set of 1, 4, 12, and 39 time steps.

Figures 4.2 and 4.3 show the general conformity of the DASCh emulation with the theoretical analysis.

Figure 4.4 shows a further regularity that was discovered experimentally, but not in the theoretical analysis. The variance of orders issued from Site 2 is inversely proportional to the square of the forecast window used to project future demand from past demand.

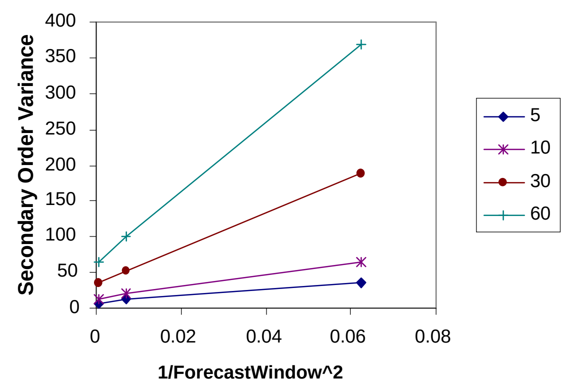

Figure 4.4: Dependence of Site 2 Order Variance on Forecast Window

Thus the experimental results confirm and extend the theoretical analysis of amplification of variance. The variance of the order stream produced by a site (the "focal site") has the general dependency

$$<\rho(t)^2> \propto L^2<\eta(t)^2>/f^2$$

where

- $<\rho(t)^2>$ is the time-averaged variance of the order stream produced by the focal site;
- L is the lead time required by the focal site to ship orders, and includes both its own cycle time and the expected time needed to deliver the order to its supplier and ship the supplier's goods back;
- $<\eta(t)^2>$ is the time-averaged variance of the order stream coming into the focal site;
- f is the forecast window over which the focal site averages incoming orders in order to generate the orders it sends to its suppliers.

Operationally, these results suggest that the focal site can minimize its contribution to the amplification of variance through the system by shortening its lead times and increasing its forecast window. The latter of these actions is of doubtful benefit, since it causes other problems, to be discussed shortly.

4.1.2. Input Networks (Lower Half of Hourglass)

Figure 4.5 shows a comparable analysis for twenty-three weeks of normalized data from our automotive OEM partner, which represents the lower (input) half of the hourglass. In contrast to the upper half, where sites must forecast demand statistically based on the history of converging customer orders, orders in the lower half diverge from a single manufacturer who has already constructed a demand projection, and so ideally can be driven by forecasts passed down the chain. The extremely high variances may be partly attributed to the fact that the upper (distribution) half of the hourglass has already amplified the natural consumer demand. In view of this high variance, the slight increase in the mean level of orders across the three levels of the supply chain is probably meaningless. As in the previous example, there is significant increase in variance from the OEM's outgoing order stream to that of the first tier supplier (in this case, more than a 3X increase). In contrast, there is virtually no amplification in variance between Tier 1 and Tier 2.

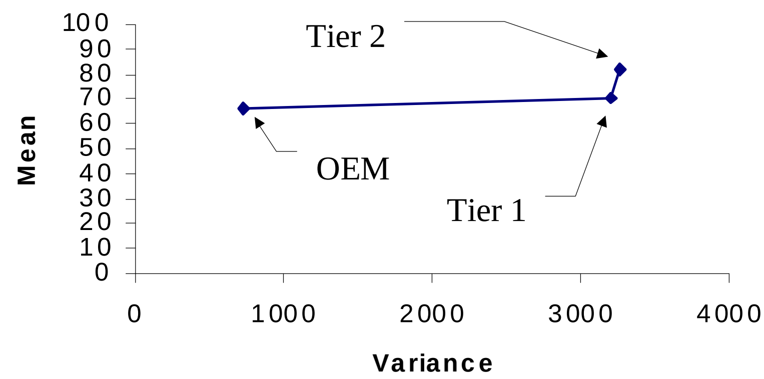

Figure 4.5: Variance Amplification in Automotive Data

We do not have a theoretical analysis of the lower half of the hourglass, but analysis of experimental data shows a distinct structure to this amplification. Now each site that receives orders amplifies the variance in the orders it sends out by the product of its lead time and any noise that it adds to the forecast. "Noise" in this case models any variation from the forecast from the focal site's customer to the forecast that the focal site provides to its supplier. Ideally, there should be no need for such variation. However, industry experience indicates that suppliers do not trust the forecasts provided by their customers, often with good cause, and as a result usually modify them before passing them on to their suppliers.

We experimented with these dynamics by varying the consumer's order variance over the set {5, 10, 20}, the noise added by Sites 2 and 3 over the set {1, 5, 15}, and the lead times of Site 2 over {6, 16, 31} and of Site 3 over {10, 20, 35}. Each configuration was run for 1000 steps, and statistics were gathered on the last 750 steps of each run to avoid any start-up transients. Under this regime, Figure 4.6 shows Site 3's outgoing orders to the Supplier plotted against the least-squares fit

$$183.7 + 3.1*(\eta(t)^2 + N_2*L_2 + N_3*L_3)$$

where

- $\eta(t)^2$ is the variance of the Consumer's order stream;

- N_2 is the variance of the noise added by Site 2,
- N_3 is the variance of the noise added by Site 3,
- L_2 is Site 2's lead time,
- L_3 is Site 3's lead time.

The two N*L terms represent amplification introduced by the sites receiving the forecast.

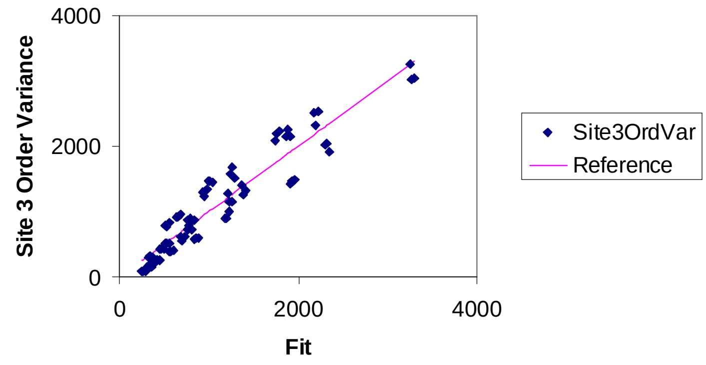

Figure 4.6: Variance Amplification in Input Networks

This model may explain the differences in amplification observed in the actual data of Figure 4.5. The Tier 1 supplier is significantly modifying the forecast from the OEM, has a long lead time, or both, and so adds a significant increment to order variance, while the Tier 2 supplier holds very close to the forecast it receives from Tier 1 or has a much shorter lead time (or both).

This analysis reemphasizes the importance of short lead times, and encourages suppliers not to modify forecasts received from their customers. Of course, the experiment assumes that the customer actually orders what the forecast predicts. If (as often happens in automotive) the forecast is inaccurate, its usefulness to suppliers decreases dramatically. In these cases, suppliers are likely to return to a weighted average forecast like that necessary in the distribution half of the hourglass, with amplification driven according to the analysis in the previous part.

4.2. Correlation of Variation

Our theoretical analysis of weighted average forecasting (Section 3.1) suggests that order processing may generate correlation in the orders seen by subtier suppliers even when the customer's original stream of orders is uncorrelated. The theory predicts that this effect will be strongest with short lead times, large variances in the incoming order stream, and a small forecast window. To test this prediction, we set up the model with Consumer orders at a mean of 100 and a variance of 50, forecast windows in Sites 2 and 3 of 5, and lead times of 2 for Site 2 and 3 for Site 3, and ran for 1000 time steps. Then we examined the results for correlation using time delay plots, in which each element in a time series is plotted on the Y-axis against the previous element on the X-axis. Figure 4.7 shows the delay plot for the Consumer orders. As expected for IID data, they form a circular blob, with no apparent structure.

Figures 4.8 and 4.9 show the orders issued by Sites 2 and 3, respectively, in response to the IID consumer orders. These plots show two interesting features. First, although plotted to the same scale, the clouds of points are larger, reflecting the amplification of variation already seen in Figure 4.1. Second, the clouds are no longer circular in shape. Now they are stretched along a line indicating X = Y. This stretching indicates that these sites are more likely to follow a large order with another large one, and a small order with another small one. In other words, their orders have become correlated in time, and increasingly so as we go deeper in the supply chain.

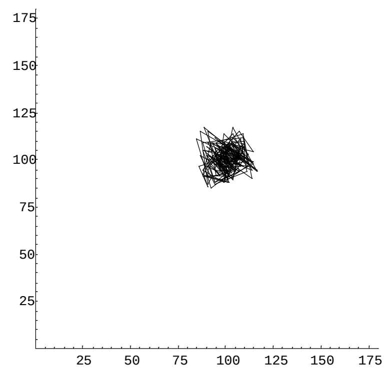

Figure 4.7: Consumer Orders

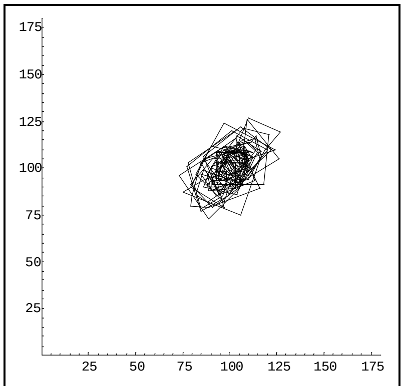

Figure 4.8: Site 2 Orders

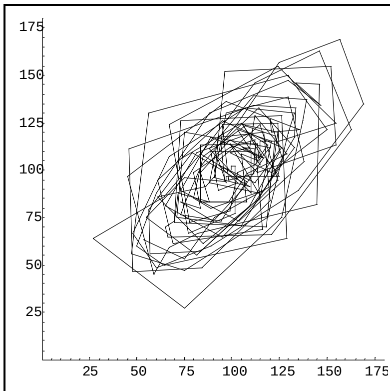

Figure 4.9: Site 3 Orders

Although our theoretical analysis of variation correlation is in the context of the distribution half of the supply network, we explored the automotive data from the input half using the same mechanism. Figures 4.10, 4.11, and 4.12 show time delay plots for the orders generated by the OEM, Tier 1, and Tier 2, respectively. The change in size of the cloud from Figure 4.10 to Figure 4.11 reflects amplification of variance due to Tier 1's order processing. The change in the shape of the cloud shows that this processing is also altering the correlation of successive orders in the stream. Because Tier 2 is passing on Tier 1's forecast virtually unchanged, Figures 4.11 and 4.12 are of the same general shape and size. The data support the conclusion that PPIC computations in both halves of the hourglass can impose spurious structure on order streams, structure that does not reflect true correlations in the demand posed by the end customer. Effective supply chain management needs mechanisms to correct for this spurious structure.

4.3. Persistence of Variation

A basic exercise in analyzing the dynamical behavior of a system is to present it with a step function. In DASCh, such a perturbation causes persistent variations in downstream ordering behavior.

Figure 4.13 shows the effect of two successive step functions in Consumer orders (the solid line) on the orders issued by Site 3 to the supplier (the dashed line), using weighted average

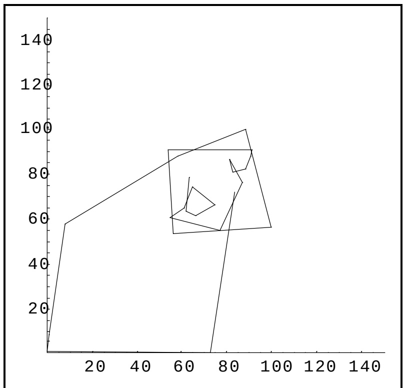

Figure 4.10: Orders from OEM to Tier 1

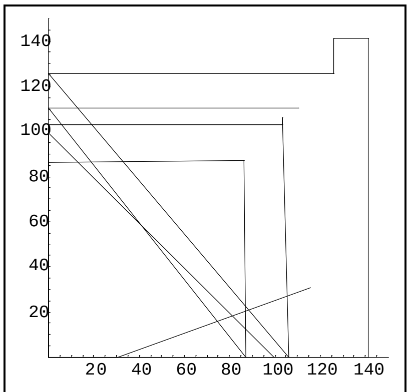

Figure 4.11: Orders from Tier 1 to Tier 2

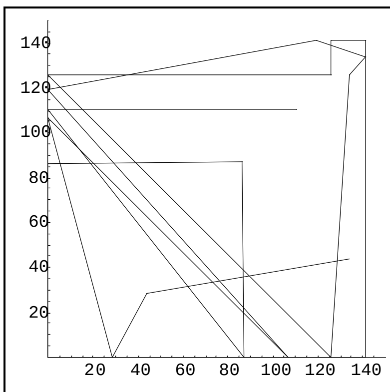

Figure 4.12: Orders from Tier 2 to Tier 3

forecasting. In both cases, the Consumer increases its order level by 10 orders per time period.

Though the change in customer orders is a one-time phenomenon, its effect persists in the orders that Site 3 issues to the Supplier. The persistence time is of the same order as the forecast window. For the first step increase in Consumer orders, the forecast window is 39 weeks and the disturbance in Site 3 orders persists for 31 weeks (to the last upward spike over the new demand level) or 47 weeks (to the downward spike). The amplitude of the variability in Site 3 orders ranges from a high of 125 to a low of 100, or a total range of 25.

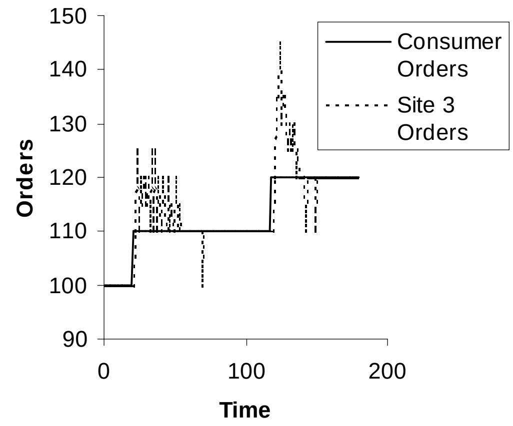

Figure 4.13: Persistence of a One-Time Disturbance

Before increasing the Consumer demand again, we cut the forecast window in both PPIC modules from 39 to 20. The period of variability lasted fewer time steps (22 to the last order above 120, or 29 to the final downward spike). But shortening the forecast window, as discussed previously, has the effect of increasing the amplification. Thus the second set of peaks is taller than the first (ranging from 110 to 145, or a total range of 35).

Thus the weighted forecasting algorithm has the effect of imposing a memory on the system. The longer the forecasting period, the longer the memory, but the lower the amplitude of the variations generated.

The implication of persistence is that supply chains have memories. They can retain the state of the chain. Forecasting windows are one such memory. Other experiments show that backlog orders and high work in process (WIP) levels also constitute implicit memories. Backlogged orders record the state of demand at the time the orders were placed, not the current demand.

These memories must be shortened to improve agility, the ability to respond rapidly to changes in the marketplace.

4.4. Generation of Variation

Now we look at a third behavior, for the first time in the nonlinear domain of our model. A chain with stable boundary conditions can generate variation internally. Assume that the customer has a steady demand with no superimposed noise. The bottom level supplier makes every shipment exactly when promised, exactly in the amount promised. Batch sizes are still 1, but now we impose a capacity threshold on sites 2 and 3: in each time step they can only process 100 parts, a threshhold nonlinearity. As long as the customer's demand is below the capacity of its suppliers, the system quickly stabilizes to constant ordering levels and inventory throughout the chain. When the top level of demand from the customer exceeds the capacity of the intervening sites, those sites see oscillation in their inventory levels. We did not expect this behavior initially, but after observing it, were able to capture it analytically in the theory set forth in Section 3.2.

Figure 4.14 shows the inventory oscillation that arises when demand exceeds capacity by 10%. Site inventories oscillate out of phase with one another, in the form of a sawtooth that rises rapidly and then drops off gradually.

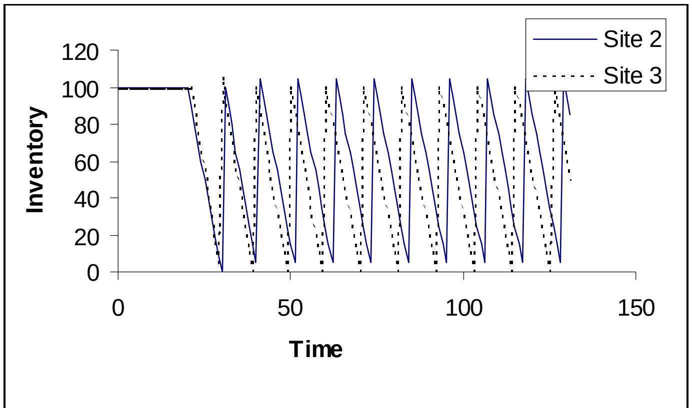

Figure 4.14: Demand/Capacity = 110/100

Detailed analysis of the experimental logs reveals the underlying mechanism behind this oscillation. The defect between an incoming order and a site's output erodes away the safety stock at that site. If a site makes 100 units per time period and is being asked for 110, it can ship 110 by taking 10 from its safety stock. On the next cycle it can still ship 110 but the safety stock drops by another 10, and so forth until the safety stock runs out. At that point, the site misses a shipment. That shipment goes into the backlog, and whatever was produced in that time period effectively generates a new safety stock. Then the process begins again. The sawtooth emerges directly from this interplay of demand, safety stock, and capacity limitation. If there is no safety stock initially, the first missed shipment will behave like a safety stock, leading to the same dynamics. The rapid rise of the sawtooth corresponds to the deposit of the site's output into inventory when it is inadequate to fill an order, while the slow fall of the sawtooth reflects the depletion of this safety stock to fill successive orders. The overall height of the sawtooth is determined by the site's capacity, and its period is determined by the degree of overload. In this case, a 100 unit base capacity supports ten time periods of 10 unit excess orders, and the missed order that replenishes the safety stock occupies another time period, resulting in a sawtooth period of eleven time units.

As the system operates above capacity, two backlogs build up: a backlog of orders waiting to be filled, and a backlog of WIP behind the capacity bottlenecks. This WIP continues to drive the site at its capacity limit even if demand subsequently drops below capacity. In this mode, each cycle produces more finished goods than are consumed by one of the new below-capacity orders. The excess goes into finished goods inventory, which builds up until it can satisfy one of the backlogged over-capacity orders, at which point the older order is filled and inventory drops in one step. The result is again a sawtooth oscillation, but of opposite direction, with gradual rise (generated by the cycle-by-cycle excess production over the new demand level) and a rapid drop (generated by filling a backlogged over-capacity order).

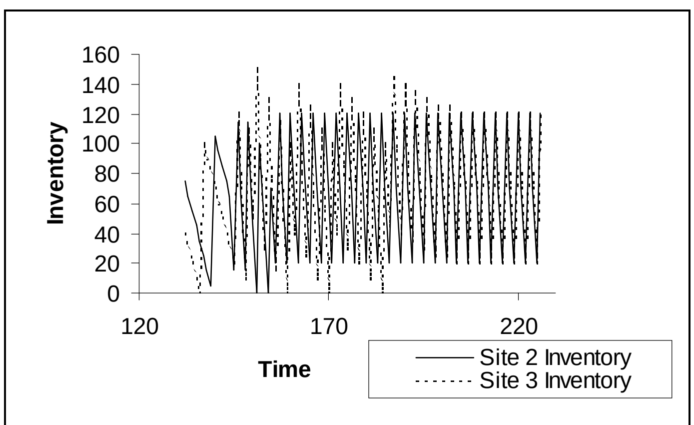

Figure 4.15: Demand/Capacity = 150/100

Figure 4.15 shows the dynamics resulting from increasing the Consumer demand from 110 to 150. After a transition period, the inventory levels settle down to a sawtooth with a shorter period. Now one cycle's production of 100 can support only two orders, leading to a period-three oscillation. The inventories of sites 2 and

3, out of synch when Demand/Capacity = 110/100, are now synchronized and in phase. (This synchronization is not explained by the theory of Section 3.2 or the abstraction of the model as a modulo function.)

The transition period is actually longer than appears from Figure 4.15. The increase from 110 to 150 takes place at time 133, but the first evidence of it in Site 2's dynamics appears at time 145. The delay in the effect is due to the backlog of over-capacity orders at the 110 level, which must be cleared before the new larger orders can be processed.

Figure 4.16 shows the result of increasing the overload even further. (Because of the increased detail in the dynamics, we show only the inventory level for Site 2.) Now the Consumer is ordering 220 units per time period. Again, backlogged orders at the previous level delay the appearance of the new dynamics; demand changes at time 228, but appears in the dynamics first at time 288, and the dynamics finally stabilize at time 300.

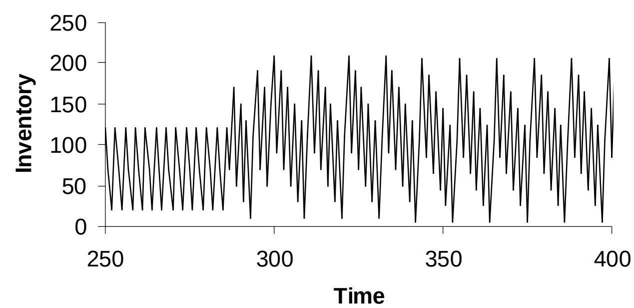

Figure 4.16: Demand/Capacity = 220/100 (Site 2)

This degree of overload generates qualitatively new dynamical behavior. Instead of a single sawtooth, the inventories at sites 2 and 3 exhibit biperiodic oscillation, a broad sawtooth with a period of eleven, modulated with a period-two oscillation. This behavior is phenomenologically similar to bifurcations observed in nonlinear systems such as the logistic map, but does not lead to chaos in our model with the parameter settings used here. The occurrence of multiple frequencies is stimulated not by the absolute difference of demand over capacity, but by their incommensurability, as detailed in Section 3.2.

4.5. Summary

The preliminary DASCh experiments show that even the simple four-stage in-line chain that we modeled supports a wide range of non-intuitive behaviors with important commercial implications. In spite of the simple nature of the model, not only our Fortune-100 informant but other industrial reviewers to whom we have showed these results recognize them as characteristic of real-world supply-chain dynamics.

5. Summary and Recommendations

This part summarizes some of the insights gained from DASCh in terms of the characteristics outlined in the project proposal, makes recommendations for supply chain operation that emerge from our research, and proposes important directions for future study.

5.1. A Dynamical Characterization of Supply Networks

The initial proposal for this effort identified four important characteristics of a supply network viewed as a dynamical system, described in terms of its state space. This section defines a model of state space for a supply network and reviews these four characteristics in terms of our experience with the model.

5.1.1. The DASCh State Space

For a given set of the parameters defined in Part 2 of this report, the major state variables of interest in DASCh are the levels of backlogged orders, finished goods inventory, and WIP inventory at sites 2 and 3. One can easily justify including many other variables in a definition of the system's overall state space, but these six variables are sufficient for the purposes of discussing the four characteristics in this section.

5.1.2. Accessibility

Accessibility describes where in its state space the system can go. We observed several constraints on accessibility in our experiments, but one important potential for restricted accessibility has not yet been observed.

The behavior of the individual site agents excludes regions of the state space in which finished goods inventory exceeds backlogged orders.

In addition, if a site's *delayMean* is 0, WIP inventory is locked at 0.

When demand is constant and less than or equal to capacity, the system settles into a point attractor whose location in state space is defined by configuration parameters, and does not visit other locations. However, the addition of noise to demand causes the trajectory to explore the vicinity of this point. Because of amplification of variance, the region that is explored is not spherical, but rather ellipsoidal, elongated along the dimensions corresponding to site 3

When demand is greater than capacity, the sites must backlog some orders. We have studied only the case when demand is constant. The amount of backlog at site 2 is an integral multiple of the customer orders. Finished goods inventory at both sites and WIP inventory at site 2 take on only a finite number of values, which depend in a complicated way on the magnitudes of demand and capacity. Specifically, finished goods inventory is attracted to the region $Production \leq Inventory(t) < Demand+Production$ (Section 3.2.2.1). Inventory can occupy any of *D* distinct values, where *D* is the numerator of *Demand/Production* reduced to lowest common terms, and never enters the bands between these values, which have width equal to *Demand/D* (Section 3.2.2.4). This behavior is typical of a limit cycle.

In both point attractors and limit cycles, the accessible volume of state space has measure zero. In our initial experiments, we have explored only a small fraction of the possible parameter combinations, and have not yet found a combination of parameters that causes the system to become formally chaotic. In such a regime the accessible volume of state space would be finite and nonzero.

5.1.3. Controllability

Controllability describes the extent to which users can steer the system trajectory in state space deliberately. Discussing this characteristic for DASCh requires us to identify the “user” of the system. Candidates are the four main sites along the backbone of the basic experimental model: the consumer, the two manufacturing sites (2 and 3), and the supplier.

The consumer exerts control on the model through the line of orders that it issues. For a given set of site parameters, the magnitude of these orders determines the point attractor or limit cycle to which the system is drawn. If the orders include noise, the attractors become less precise, in ways not directly under the control of the consumer. That is, the amplification, correlation, and persistence phenomena, driven by the PPIC mechanisms, limit the consumer’s control over chain-wide state variables.

The sites exert control on the model through the parameter settings that they select for themselves and their associated PPIC agents (as outlined in Tables 2.2 and 2.3). In Part 5, we have described some of the trade-offs in these parameters. For example, a long forecast window reduces amplification but increases persistence. Based on our experience thus far, it appears likely that improved controllability from the standpoint of manufacturing sites depends not on finding the “right” parameters, but on developing adaptive behaviors for the sites.

5.1.4. Inertia

Inertia measures how fast the system can move from one state to another.

In our experiments, DASCh manifests inertia in two ways. First, the persistence phenomenon is a form of inertia that is driven by the forecast window: the longer the forecast window, the more inertia in the system, and the longer it takes to forget past state. Second, when demand exceeds capacity, unsatisfied orders build up at the sites, along with WIP at site 3, and subsequent changes to the level of demand are not visible until previous orders have been satisfied. The greater the excess of demand over capacity, the longer it takes the producers to work through the backlog and WIP, and the greater the inertia.

5.1.5. Performance

Performance measures how well the system performs at a fixed location in its state space. The main performance measures we have used in these experiments are the magnitudes of finished-goods and WIP inventory, which are direct values of state variables. The model supports measurements of throughput at manufacturing sites and average time to fill and average lateness at the consumer, but the current set of experiments has focused on understanding the basic dynamics of the system rather than optimizing these performance metrics.

5.2. Operational Recommendations

The experimental results in the previous part of this report suggest a number of specific operational recommendations that concern two aspects of system management: PPIC and overall system scaling.

5.2.1. PPIC

Most of the anomalous behavior we detected can be traced to the PPIC agent, which in our model implements a simple version of the widely-used MRP algorithm. This agent embodies the only adaptive behavior currently assigned to the manufacturing sites, and it is effective in reducing variability in average time to fill orders at the consumer. However, companies that use such an algorithm in managing their order stream should be aware of some of the anomalies to which it can lead:

- Any forecasting of demand based on past orders will lead to some amplification of variance [16]. Long lead times and distortions of customer forecasts exacerbate this effect, particularly when they are combined at the same site. *Recommendation*: Lead time reduction has long been recognized as an important discipline in manufacturing improvement; its contribution to variance amplification is an additional and previously unrecognized motive to pursue such activities. The contribution of distortions of customer forecasts suggests that members of the supply network resist the temptation to adjust such forecasts. This recommendation applies not only to suppliers, but also to OEM's, who too often distort their internal forecasts as a deliberate effort to manipulate supplier behavior (for example, to lock in supplier capacity). Experience with the DASCh model led our Fortune-100 informant to restrict the manipulations to forecasts traditionally made within his company, and he traces subsequent performance improvements in inventory levels to this change in behavior.
- While longer forecasting windows reduce amplification, they are a major component of inertia, leading to persistence of the effect of a change in demand. Other forms of memory (backlogged orders and WIP) can lead to a different variety of inertia: a delay in the time at which one becomes aware of a change in external conditions. *Recommendation*: Recognize the tension between amplification and inertia, and take both explicitly into account in setting forecasting windows.
- PPIC imposes structure on the demand stream, described in Part 5 as "correlation." This structure is spurious in the sense that it does not reflect top-level requirements, but is purely an artifact of the dynamics of the system. *Recommendation*: Low-level suppliers should be extremely suspicious of systematic variability that they see in their incoming order stream, and should confirm it independently (for example, through information received directly from the OEM) wherever possible before making business decisions.
- One important input to PPIC is the current inventory level. We have found that this level can be driven into complex oscillations by excess of demand over capacity. Current PPIC algorithms do not take these oscillations into account. While we have not analyzed in detail the implications of such a driving function on PPIC behavior, they can hardly be negligible. *Recommendation:* Variations in inventory should be reviewed for possible origins in mismatch of demand to capacity before being used directly as input to PPIC computations. The patterns identified in Section 3.2.2 may be useful in constructing such diagnostics.

5.2.2. System Scaling

The oscillatory behavior we observed shows how a bottleneck in the supply chain not only limits capacity, but also introduces variation. This behavior leads to two recommendations:

- *Recommendation*: Leanness has its limits. Suppliers should be careful to provide adequate capacity for expected demands, and should plan for the consequences of undertaking commitments in excess of that capacity.
- *Recommendation*: OEM's should recognize the possible performance and long-run cost consequences of overloading suppliers (for example, by adding safety buffers to early orders and canceling later ones to compensate for any over-deliveries).

5.3. Future Research

The unexpected results found in this study offer a rich foundation for data-analytic study of real firms, and only scratch the surface of the exploration that can usefully be done with the present model. In addition, it is desirable to extend the structure of this model to match a wider range of realistic supply networks, and to explore more sophisticated mechanisms for overcoming the undesirable behaviors that we have identified.

5.3.1. Data-Analytic Opportunities

The current project has focused on constructing and exploring a simple model of a manufacturing supply chain. To date, we have ensured the realism of our model qualitatively, by close collaboration and regular review with a manufacturing manager at a major electronics firm. The results in hand (such as the different functional dependencies of amplification in distribution vs. input networks, or the detailed structure of capacity-induced oscillations) suggest signatures for which real-life inventory data could be tested. Such comparison will not only provide a quantitative measure of the fidelity of the model, but also suggest the most appropriate enhancements to reflect more complex industrial practices than those we have currently modeled.

5.3.2. Current Structure and Algorithms

The current model will support extended experimentation in three directions.

1. What is the effect of introducing variation (either structured or noise) at different points in the system? We have systematically explored only a few of the parameters that are equipped with noise modification, and have varied demand only as an isolated step function. It would be interesting to explore (for example) the interaction of real periodic variations in demand with the correlations generated by PPIC, or with inventory oscillations induced through capacity limitations. If we drive the system in its oscillatory parameter regime with demand that oscillates at some incommensurate frequency, we should see something like quasi-periodicity. There are also parameter regimes in which quasi-periodicity becomes chaotic, so we might well be able to produce chaotic behavior in this system. We need to see also whether such parameter regimes are reasonable for operating supply chains.

2. What dynamics emerge from coupling the two sides of the supply network hourglass together? Our experiments to date use the same forecasting mechanism at all sites. By constructing a longer chain, we can reasonably use weighted forecasting in the upper half and prediction in the lower half.

3. How does the behavior of the network vary with the scale of production? We expect the dynamics of large-lot production (cars and toasters), which we have explored most systematically, to differ significantly from small-lot production (ships and aircraft). The latter domain will be important for understanding the effects of mass-customization. (It may be that useful results for small-scale systems will require structural extensions described in the next section.)

5.3.3. Structural Extensions

The current model supports only linear sequences of sites. The model should be extended to support arbitrary networks. Such an extension will permit us to model three important real-world situations:

1. The effects of assembly (a single site with multiple suppliers) and disassembly (a single site with multiple consumers). This application requires significant modifications to PPIC.

2. Multiple products with common components (for example, several models of computer, all using the same power supply). This application requires that sites be able to allocate components across products, presumably on the basis of production quotas.

3. Competitive suppliers and/or consumers. This application also requires sites to make allocation decisions, presumably in this case on the basis of cost and price information.

5.3.4. Adaptive Mechanisms

The current work has shown the limitations of traditional PPIC as an adaptive mechanism for individual sites. Several approaches deserve exploration for their potential to control the undesirable behaviors we have identified.

1. The forecasting techniques we modeled, while representative of those used in industry, are primitive. We intend to exploit more sophisticated methods that draw on the characteristics of the data stream.
2. The insights gained in this research offer a rich new perspective from which to evaluate PPIC algorithms. The behavior of common algorithms in the face of demand-driven oscillations or persistence from past unrepeated events needs to be understood much better. It is likely that significant improvements will be necessary to cope with the resulting distortions.
3. Forecasting is one form of learning, focused on one aspect of the environment (the incoming order stream). Learning might also be usefully applied to the performance of suppliers, in compensating for deviations between orders and shipments. Especially in a nonlinear network with competing suppliers, a site might vary its preference for one supplier over another based on past performance. Similarly, with multiple customers for a scarce product, a site might shift its preference from one to another on the basis of past accuracy of predictions.
4. Currently, the only signals providing coordination among sites in our model are orders, shipments, and (in the lower half of the hourglass) predictions of future orders. Supplementing such signals with market mechanisms on which the persistence of agents depends has proven useful in other domains [11] in controlling disorder due to nonlinearity. Extension of the structure to permit competitive supply and consumption will allow us to

explore the impact of dynamic bidding mechanisms, in which the flows of orders and shipments change over time based on bidding across competitors.

5. As forecasting, learning, and coordination mechanisms become more sophisticated, the parameters that one can adjust in a site proliferate beyond the limit of manual exploration. We expect the use of genetic mechanisms to explore parameter combinations to be both necessary for manageable experimentation and promising for the development of greatly improved performance.

References

[1] E. G. Anderson, Jr., C. H. Fine, and G. G. Parker. Upstream Volatility in the Supply Chain: The Machine Tool Industry as a Case Study. http://imvp.mit.edu/imvpfree/Fine/upstream.pdf, MIT Sloan School, Cambridge, MA, 1996.

[2] R. Axtell. Three Distinct Uses of Agent-Based Computational Models in the Social Sciences. Brookings Institution, Washington, DC, 1997.

[3] J. L. Burbidge. Automated Production Control with a Simulation Capacity. In P. Falster and R. B. Mazumder, Editors, *Modelling Production Management Systems*, pages 19-35. Elsevier Science Publishers (North-Holland), 1985.

[4] J. F. Burns and B. D. Sivazlian. Dynamic Analysis of Multi-Echelon Supply Systems. *Computers and Industrial Engineering*, 2:181-193, 1978.

[5] G. P. Cachon. On the Value of Managing Supply Chain Demand Variability with Scheduled Ordering Policies. http://www.duke.edu/~gpc/..., Duke University, Durham, NC, 1997.

[6] B. Eberlein. System Dynamics Mailing List. 1997. Available at http://www.vensim.com/SDMAIL.HTM.

[7] M. E. Epstein and R. Axtell. *Growing Artificial Societies: Social Science from the Ground Up*. Boston, MA, MIT Press, 1996.

[8] P. A. Fishwick. *Simulation Model Design and Execution: Building Digital Worlds*. Englewood Cliffs, NJ, Prentice Hall, 1995.

[9] J. W. Forrester. *Industrial Dynamics*. Cambridge, MA, MIT Press, 1961.

[10] High Performance Systems. ithink - The Premiere Business Simulation Tool from High Performance Systems, Inc. 1997. Available at http://www.hps-inc.com/products/ithink/ithink.html.

[11] T. Hogg and B. A. Huberman. Controlling Chaos in Distributed Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6 (November/December)):1325-1332, 1991.

[12] K. R. Howard. Unjamming Traffic with Computers. *Scientific American*, (October), 1997.

[13] O. P. Judson. The Rise of the Individual-Based Model in Ecology. *Trends in Ecology and Evolution*, 9(1 (January 1994)):9-14, 1994.

[14] G. A. Kruger. The Supply Chain Approach to Planning and Procurement Management. *Hewlett Packard Journal*, (February):6.1-6.9, 1997.

[15] H. L. Lee and C. Billington. Material Management in Decentralized Supply Chains. *Operations Research*, 41(5 (September-October)):835-847, 1993.

[16] H. L. Lee, V. Padmanabhan, and S. Whang. Information Distortion in a Supply Chain: The Bullwhip Effect. *Management Science*, 43(4 (April)):546-558, 1997.

[17] F.-R. Lin, G. W. Tan, and M. J. Shaw. Multi-Agent Enterprise Modeling. Office of Research, College of Commerce and Business Administration, University of Illinois, Urbana, IL, 1996.

[18] H. V. D. Parunak. The DASCh Experience: How to Model a Supply Chain. In *Proceedings of Second International Conference on Complex Systems*, 1998.

[19] H. V. D. Parunak, R. Savit, and R. L. Riolo. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In *Proceedings of Workshop on Multi-agent systems and Agent-based Simulation (MABS'98)*, Springer, 1998. Available at http://www.erim.org/~van/mabs98.pdf.

[20] F. Reif. *Fundamentals of Statistical and Thermal Physics*. New York, McGraw-Hill, 1965.

[21] G. P. Richardson and A. L. Pugh. *Introduction to System Dynamics Modeling with Dynamo*. Cambridge, MA, MIT Press, 1981.

[22] H. A. Simon. On the Application of Servomechanism Theory in the Study of Production Control. *Econometrica*, 20:247-268, 1952.

[23] J. D. Sterman. Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment. *Management Science*, 35(3 (March)):321-339, 1989.

[24] T. J. Strader, F.-R. Lin, and M. J. Shaw. Simulation of Order Fulfillment in Divergent Assembly Supply Chains. *Journal of Artificial Societies and Social Simulation*, 1(2), 1998. Available at http://www.soc.surrey.ac.uk/JASSS/1/2/5.html.

[25] J. M. Swaminathan, N. M. Sadeh, and S. F. Smith. Sharing Supplier Available Capacity Information. Carnegie Mellon University, Pittsburgh, PA, 1997.

[26] J. M. Swaminathan, S. F. Smith, and N. M. Sadeh. Modeling Supply Chain Dynamics: A Multiagent Approach. *Decision Sciences*, forthcoming, 1997.

[27] D. R. Towill. Dynamic analysis of an inventory and order based production control system. *International Journal of Production Research*, 20(6):671-687, 1982.

[28] D. R. Towill. Supply Chain Dynamics. *International Journal of Computer Integrated Manufacturing*, 4(4):197-208, 1991.

[29] D. R. Towill, M. M. Naim, and J. Wikner. Industrial Dynamics Simulation Models in the Design of Supply Chains. *International Journal of Physical Distribution and Logistics Management*, 22(5):3-13, 1992.

[30] S. Tzafestas and G. Kapsiotis. Coordinated control of manufacturing/supply chains using multi-level techniques. *Computer Integrated Manufacturing Systems*, 7(3):206-212, 1994.

[31] F. Villa. New Computer Architectures as Tools for Ecological Thought. *Trends in Ecology and Evolution*, 7(6 (June 1992)):179-183, 1992.

[32] M. E. Warkentin. MRP and JIT: Teaching the Dynamics of Information Flows and Material Flows with System Dynamics Modeling. In *Proceedings of The 1985 International Conference of the Systems Dynamics Society*, pages 1017-1028, International System Dynamics Society, 1985.

[33] J. Wikner, D. R. Towill, and M. Naim. Smoothing supply chain dynamics. *International Journal of Production Economics*, 22:231-248, 1991.

[34] W. G. Wilson. Resolving Discrepancies between Deterministic Population Models and Individual-Based Simulations. *American Naturalist*, 151(2):116-134, 1998.

