Information Visibility And Its Effect On Supply Chain Dynamics

by

Yogesh V. Joshi

B. Tech., Mechanical Engineering (1998)
Indian Institute of Technology, Bombay, India

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the Degree of

Master of Science

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2000

Signature of Author ....
Department of Mechanical Engineering
May 18, 2000

Certified by ....
Sanjay Sarma
The Cecil and Ida Green Career Development Chair
Associate Professor for Mechanical Engineering

Accepted by ....
Ain A. Sonin
Chairman, Department Committee on Graduate Students

Information Visibility And Its Effect On Supply Chain Dynamics

by

Yogesh V. Joshi

Submitted to the Department of Mechanical Engineering
on May 8, 2000 in partial fulfillment of the
requirements for the Degree of Master of Science

Abstract

Supply chains are nonlinear dynamic systems, the control of which is complicated by long, variable delays in product and information flows. In this thesis, we present a novel framework for improving the visibility of information in supply chains by reducing the delays in information flow. We first analyze the growth and evolution of production and operations management software over the past three decades, and the current trends in their development, coupled with recent advances in radio frequency technology, wireless communications, data representation methods, and the internet. Information visibility is identified as one of the key elements for successful implementation of any such software. We analyze the dynamics of a supply chain under different scenarios of information visibility and forecasting decisions with the help of simulations. Possible improvements in supply chain costs are identified, provided information visibility. We propose a framework to achieve information visibility in the supply chain using radio frequency tags, tag readers, product identification codes, an object description language, and the internet.

Thesis Supervisor: Sanjay Sarma
Title: The Cecil and Ida Green Career Development Chair
Associate Professor for Mechanical Engineering

Acknowledgements

Many people have contributed to my research for completing this thesis. Its been a wonderful learning experience.

I am grateful to Sanjay for providing me the opportunity and resources to work on a project as exciting as this. Its been great working with Sanjay – he is very encouraging, and as Prasad once told me, his enthusiasm for work is highly contagious. My numerous brainstorming sessions with him have always greatly revitalized my motivation.

I'd like to thank all at the Auto-ID Center for their support. Special thanks to Niranjan for the numerous debates and discussions I've had with him about every conceivable topic under the sun.

My thanks to all at RAMLab for their help – especially Mahadevan whom I inevitably call whenever I have a question and don't know whom to ask; and Paula for bearing so many of those zwrites.

Thanks to David Rodriguera for making my life at MIT simple!

And a final big thanks to my parents and all my friends @ MIT and in Cambridge/Boston (I dare not start listing names lest I skip one!) for their encouragement and support!

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

Manufacturing environments today have lean machines and optimized manufacturing processes. While a great deal of research is underway to improve efficiencies with these strategies, manufacturing companies are also looking for better ways of getting their products to customers in a timely and cost-efficient fashion – *i.e.*, for managing their complex, intricate supply chains. A recent survey [1] indicates that on an average, most manufacturing companies spend about 12% to 15% of their revenues on supply chain activities. An improved manufacturing process might fail to achieve benefit for the company if the product manufactured is not selling in the market, or is not made available in the right place at the right time. A situation wherein a customer goes to a retail store to buy a particular product, finds that product to be out of stock, and ends up buying a competitor's product, is commonly reported by companies such as Procter & Gamble. Such incidents lead to lost sales and revenues for the company in question. The marginal cost of opportunity of lost sales is more for the company than for the retailer. (The retailer doesn't lose much if he runs out of stock for detergent A, since most likely the customer will go ahead and buy a detergent B.) At the same time, excessive stock raises inventory costs and likelihood of the product getting outdated. Shortening of production cycles makes it more and more important that the product gets to the market rapidly. If the production cycle is shorter than the time a product spends in the supply chain, the product will go out of production before the customer receives it. This makes it impossible to detect and react to quality problems. This follows from Little's law, which states, "the average number of jobs in the system, L, is given by

$$L = \lambda W \qquad (1.1)$$

where, λ = the average arrival rate of jobs; and

W = the average time a job spends in the system."

From Little's law, its clear that other things equal, the average time that a product spends in a supply chain is directly proportional to the average number of jobs in the system. Due to abovementioned reasons, companies feel a need for visibility of appropriate information into the product distribution network, and real time information updates to optimize scheduling and planning costs.

Today, companies use a variety of software applications for obtaining appropriate product information, and for planning and optimizing performance of their supply chains. These applications include Materials Requirement Planning (MRP), Manufacturing Resources Planning (MRPII), and Enterprise Resource Planning (ERP) for their day-to-day planning activities. Some companies have taken initiative to implement Advanced Planning and Scheduling (APS) packages for simultaneous scheduling and planning with their supplier/s. Most of these applications are focused internally within an enterprise. They depend on data gathered at regular intervals from purchasing, manufacturing, distribution, and sales operations. Current techniques for gathering data include manual

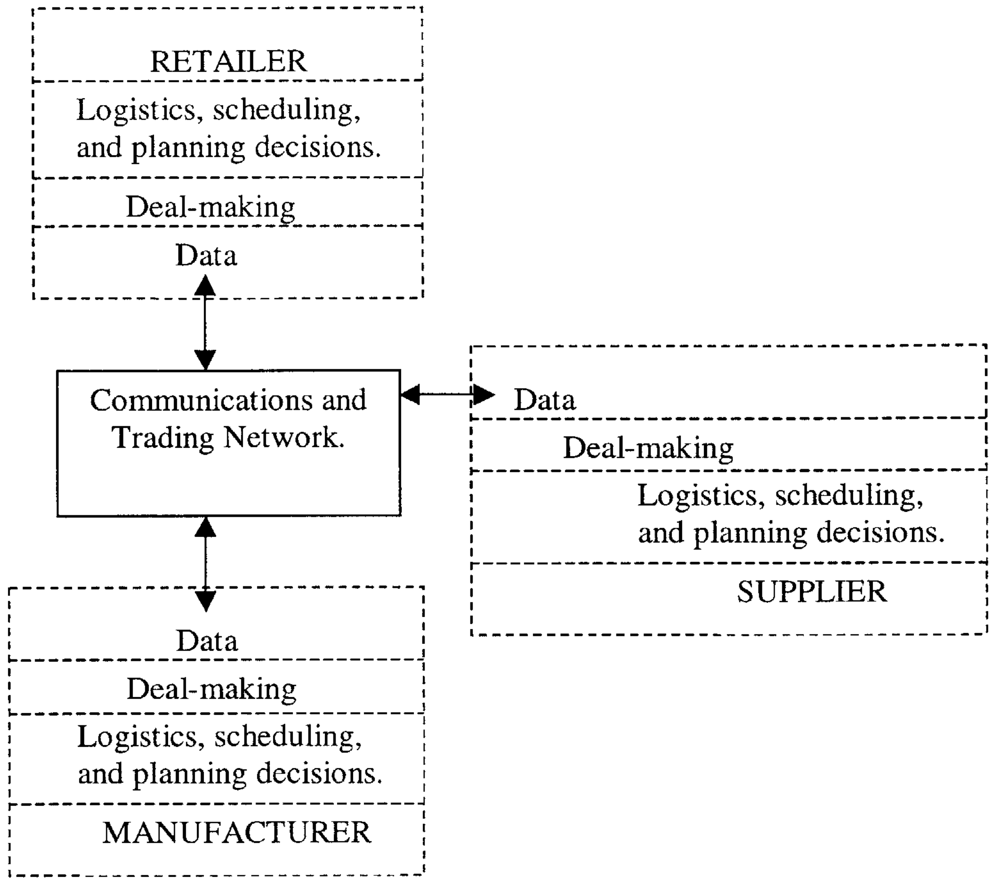


Figure 1.1 Dynamic Communication and Trading.

data entry by operators at various locations into logbooks or data-entry terminals, usage of barcodes, and tags. Transfer of information across units is usually effected by mail, phone, fax, email, or electronic data interchange (EDI). Data gathered by these techniques is typically not in real time – it is updated on a daily, weekly or monthly basis. Also, there is significant percentage of errors in manual data entry. Barcode scanning requires operators and introduces constraints regarding orientation of the product and cleanliness of labels for fast, efficient data collection. Using EDI is expensive, and moreover, not all suppliers and buyers have the infrastructure setup to use it. As a result, information access is usually restricted to localized zones. Communication between units is normally pipelined sequentially, and revising and reorganizing of plans takes a considerable amount of time. Differing data formats across trading partners introduce incompatibility – and a need to convert data from one format to another.

The advent of the internet has led to the emergence of new business models and increased competitive pressures, forcing companies to operate more efficiently than ever before. To be profitable and to thrive, companies are collaborating closely with all partners in the supply chain – from the supplier's supplier to the customer's customer. These trading partners need to share forecasts, manage inventories, schedule labor, optimize deliveries, and thus improve overall productivity. Software for Business Process Optimization (BPO), and Collaborative Planning, Forecasting and Replenishment

(CPFR), are correspondingly evolving to help companies collaboratively forecast and plan amongst partners, manage customer relations, and improve product life cycles and maintenance. Traditional supply chains are rapidly evolving into "dynamic trading networks" [2] – comprised of groups of independent business units sharing 'planning and execution information' to satisfy demand with an immediate, coordinated response. This is shown in Figure 1.1.

Most of the software packages mentioned above fall in the realm of 'Logistics, Scheduling and Planning' shown in Figure 1.1. The second layer, *i.e.*, deal-making amongst trading partners, is dependent to a huge extent on company strategies and proper alignment of business motives. Moreover, both these layers rely on the communications management layer for their data acquisition and storage needs. Thus, to facilitate interaction between these partners in a supply chain or independent business units in a dynamic trading network, it is essential to establish a strong communications link that is capable of gathering information in real-time and making it available to everyone concerned instantaneously, preferably in a standardized format. Information gathered is useful not only for collaboration amongst units but also for planning and scheduling within a unit – based on data inputs from the same unit as well as other units. For example, to decide the production schedule in an assembly plant, a car manufacturer needs information about inventories at the distribution centers and retailers, a unified forecast of demand for the cars, capability of suppliers to provide required parts for assembly, as well as current capabilities of the assembly plant under consideration, in terms of inventory levels, labor, scheduled shutdowns, etc. In this thesis, we provide a framework for achieving complete information visibility in supply chains or trading networks using the internet, and technology being developed at MIT's Auto-ID Center – electronically coded tags, automatic identification systems, and standardized formats for data representation.

The thesis is laid out as follows: In Chapter 2, we review the basics of supply chains and their changing needs with recent changes in business models, development of the internet, and rise of new software applications. We analyze the current practices in supply chain management (SCM) – the evolution of enterprise applications (MRP, MRPII, ERP) to inter-enterprise applications (APS, BPO, CPFR). In Chapter 3, we demonstrate the need for visibility in supply chains, and with help of simulations, the benefits achieved by improving visibility. In Chapter 4, we propose a framework for achieving complete visibility, and concretizing instant communication amongst all units in the supply chain or trading network. In Chapter 5, we conclude by illustrating the benefits and improvements provided by implementing the proposed framework; and propose issues to be researched further for seamlessly implementing and integrating the framework with simultaneous developments in associated fields.

Chapter 2

Analysis of current SCM practices

2.1 Introduction

The past three decades have seen tremendous developments in software for production and operations management. This has been a direct result of developments in computers and information technology, and also of the way companies ran their businesses. There has been a distinct trend of increased collaboration within organizations and amongst trading partners over the years. The traditional way of doing business with clear-cut lines of demarcation between roles and responsibilities of individual units is fast giving way to shared roles and responsibilities amongst trading partners. The growth of software applications over the years reflects these trends.

The foremost software application developed during the early 1970s to help warehouses to plan inventory and shop floors to plan production was materials requirement planning (MRP). The widespread popularity of MRP in manufacturing departments prompted the development of manufacturing resource planning (MRP II) in the 1970s. MRP II built upon MRP, by tying it to the company's financial system. By late 1980s, companies found an increasing need to integrate together information from all different units within the organization in order to be able to take better decisions for improving productivity and increasing profits. This led to the development of enterprise resource planning (ERP) applications. ERP built upon MRP II, by adding functionality to include many more departments within the organization. Implementation of ERP involved extensive use of developments in information technology. With competition increasing with time, to remain in business, companies soon found it necessary to optimize the entire product "supply chain". This called for collaboration not only within the organization, but also with trading partners in the supply chain. The importance of managing customer relationships, being flexible to respond to changes in organizational structure as well as customer demand, managing the product life cycle, etc. influenced the growth of next generation software applications - advanced planning and scheduling (APS). This software used optimization algorithms to compute the optimal production plan and machine schedules to reduce operating costs and improve profits. Competition, partner collaboration and increasing demands for customer responsiveness drove further developments in APS, and these newer software packages, generically known as Business Process Optimization (BPO) software are slowly replacing ERP/MRP II/MRP across all industries [3]. Having optimized so far, companies now are looking for ways to reduce lost sales, match supply and demand with least inventory, and remain as lean as possible. Companies are also adapting a new concept called collaborative forecasting, planning and replenishment (CPFR) to achieve the above.

In this chapter, we review the developments from MRP to CPFR. We analyze the motivation, structure, examples, benefits and drawbacks for each of these applications.

2.2 Materials requirement planning (MRP)

In a manufacturing operation, answering questions regarding which materials and components are needed, in what quantities, and when, is extremely vital. Traditionally, majority of the manufacturing organizations controlled sub-assemblies and components using order-point methods. In the early 1970s, a software application was developed to provide companies with answers to the above questions – it was materials requirement planning (MRP). An MRP system uses as inputs the demand information from the master production schedule (MPS) with a description of what components go into a finished product (the bill of materials - BOM), the order or production times for components, and the current inventory status. The system calculates the exact quantity, need date and planned order release date for each of the sub-assemblies, components and materials

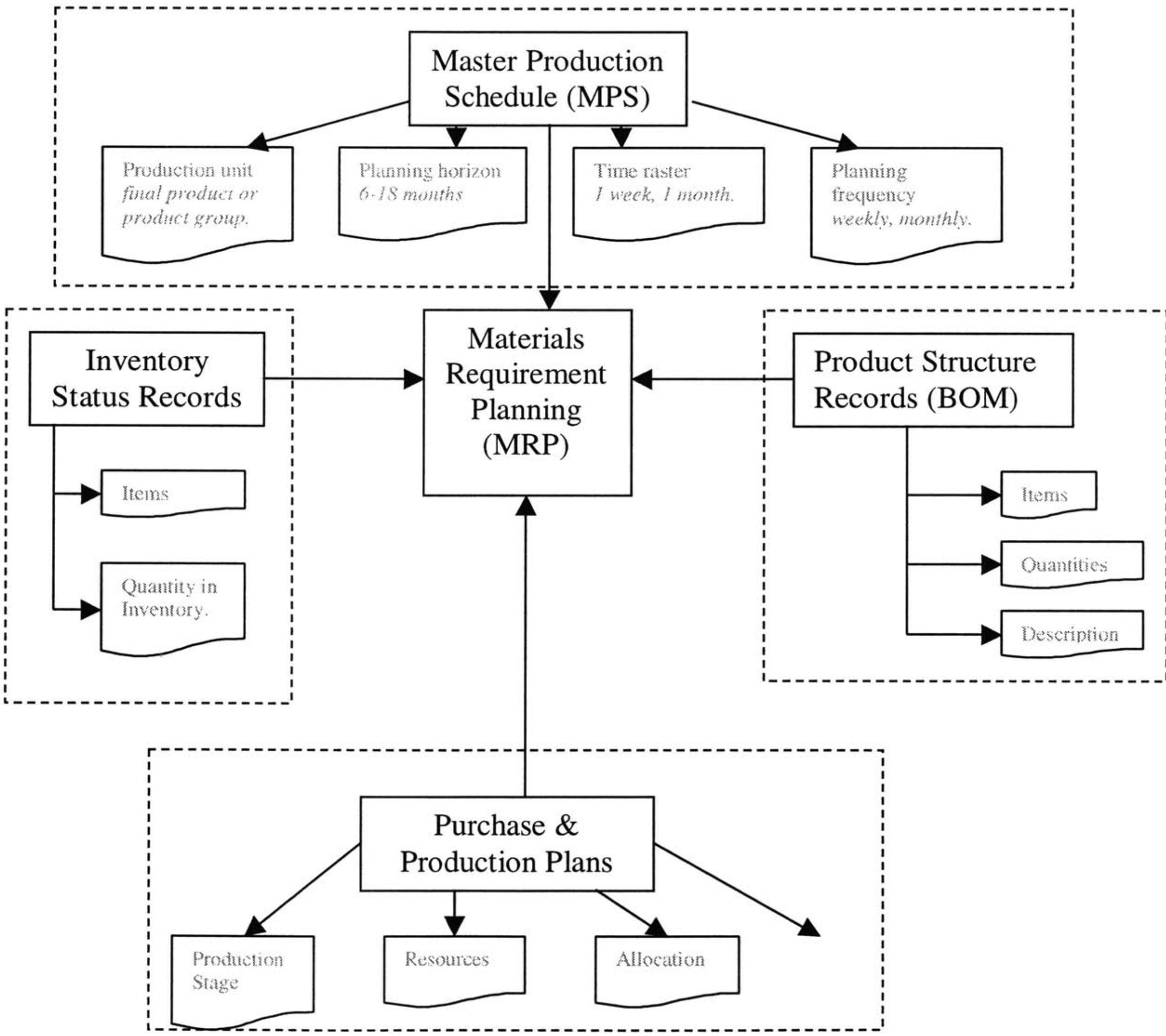


Figure 2.1 MRP: inputs & outputs.

required to manufacture products listed on the MPS [4]. This is shown in Figure 2.1. MRP aids in controlling inventories, and managing work orders, purchase orders, and sales orders. MRP helps companies adapt easily to changes in customer requirements by revising production and purchase plans when the MPS is changed. It improves customer service by providing ability to the company of consistently delivering finished products to the customer in a reliable and timely fashion. It helps maximize resource utilization, and reduce costs by pruning inventory to minimum required levels. Table 2.1 lists some examples of commercial MRP software packages.

Implementing and operating an MRP system was a major challenge for many companies. The program makes assumptions like infinite capacity, certain economic batch quantities, and fixed lead times. MRP success requires a realistic master production schedule, methods of controlling as well as planning priority, accurate purchasing lead times, a balanced approach to processing change (handling unplanned events), and most importantly, accurate data and timely data processing.

MAS 90 MRP Module for Windows	(http://www.20-20.com/MAS90_MRP.htm)
MAX ML System	(http://www.cimco-th.com/micromax.htm)
TXbase Manufacturing System	(www.txbase.com)
CMS Manufacturing System	(http://www.cybor.co.nz/cms/cmsmanu.html)
INMASS MRP System	(http://www.inmass.com/)
Concorde XAL System	(http://www.idsc.co.uk/idp/xal/m&r.htm)

Table 2.1 MRP Software

At the base of all MRP computations are correct BOM and inventory status records. If these are inaccurate, the MRP system will plan wrong items and wrong quantities – garbage in, garbage out. The salient problems arising from inaccurate records are the lack of required items, disrupted production, and late deliveries. Data used by MRP is taken from the records maintained by different units in the company. The methods used for data acquisition and records keeping are reviewed later in this chapter.

Companies soon realized that information in an MRP system is useful to functional areas other than manufacturing and operations. By late 1970s, a new policy emerged from MRP. It evolved into Manufacturing Resource Planning, or MRP II.

2.3 Manufacturing resource planning (MRP II)

The manufacturing modules of MRP II include all the elements of MRP, plus additional developments:

- Feedback – from the shop floor as to how the work has progressed, to all levels of the schedule, so that the next run can be updated on a regular basis.
- Resource scheduling – it takes into account the plant and equipment required to convert raw materials into finished goods while scheduling. This means that capacity is taken into account (unlike MRP). The drawback is that capacity is

considered only after the MRP schedule has been prepared; hence time allotment might be insufficient.

- Batching rules can be incorporated into the scheduling of resources, either Lot for Lot, economic batch quantity, or part period cover rules [5].
- Other software modules like "rough cut capacity planning" (RCCP) can be incorporated to help the scheduling process.

The focus of MRP II is to aid better management of company resources by providing information based on the production plan to all functional areas or units. It enables testing of "what if" scenarios by using simulation [6]. For example, a shop manager can see the effect of changing the MPS on the purchasing requirements for certain critical suppliers or the workload on bottleneck work centers, without actually authorizing the schedule. The MRP plan can be used along with prices and product and activity costs from the accounting system to project dollar amounts for shipments, product costs, overhead allocations, inventories, backlogs, and much more. Information from MRP II is useful to many functional areas in the firm as follows:

- Purchasing – purchase orders
- Production – production scheduling and control, inventory control, capacity planning
- Finance – financial resources needed for material, labor, overhead, etc
- Accounting – actual cash flow projections over time, production costs, etc

Although planning information generated by an MRP II system provides insights into the implications of MPS and materials plan, it has its drawbacks. The system works off fixed lead times, and does not allow for variable lead times, a very unrealistic assumption. In MRP II too, batch-sizing rules are fixed, and batch splitting is discouraged, unless "expediting" a batch to speed up a late delivery.

Since a full MRP II implementation can act as an integrated database for the company, it means that the company must put great emphasis on data accuracy. Errors in recording in one part of the system will result in problems for all users. Suppliers of MRP II software encourage users to aim for a data accuracy of between 95 to 98% [5]. Methods used for data acquisition and records keeping are discussed later in this chapter.

MRP II focuses on internal operations in a company. The system was devised to cater for the situation where most manufacturing is carried out at a single site, with many outdated assumptions. Soon the need was felt for providing and exchanging information directly with other functional units and manufacturing sites within the enterprise. To incorporate these requirements, a new system called "enterprise resource planning" (ERP) has evolved from MRP II.

2.4 Enterprise resource planning (ERP)

The development of ERP systems was an inside-out process of evolution, starting from inventory control packages, to MRP to MRP II, further expanding to include other enterprise processes such as sales and order management, marketing, purchasing, warehouse management, financial and managerial accounting, and human resources management. While MRP II includes many of these functions, it looks inwards at the heart of individual sites whereas ERP looks out to the wider picture at the entire enterprise level.

An ERP system is a configurable information systems package that integrates information and information based processes within and across functional areas and multiple sites in an organization [7]. ERP represents the application of new information technology to the MRP II model. This includes the move to relational database management systems (RDBMS), the use of a graphical use interface (GUI), open systems, and client-server architecture.

ERP systems are developed based on a reference enterprise business model, chosen by the developers of the ERP system. The developers implicitly promote the notion that the reference model used embodies best business practices. Different reference models reflect different preferred business practices, including underlying data and process models as well as organizational structures. There can be considerable mismatches between the actual company-specific business practices and the reference models embedded in the ERP system. While at the abstract level best practices may be "universal", at the detailed process level these mismatches create considerable implementation and adaptation problems.

Mismatches can also occur between the actual organizational structure and the implicitly embedded organizational structure in the reference model of the ERP software. Most of the current generation of ERP packages are based on a traditional hierarchical, functional view of the organization. Work in organizations can be distributed over many geographic and/or organizationally dispersed regions. An organization can be global – ignoring national boundaries, structured by functional units; or it could be multinational, wherein, administrative structure is demarcated into convenient geographical units, with local regulations and capabilities. It can support multiple languages and currencies, or enforce use of a single language and currency, or any combination thereof.

Visibility of transactions across units can be at a detailed or aggregate level. Ownership of data can be centralized and owned by the company or localized and owned by individual units. The organization can choose between using multiple database architecture, single database architecture, and client-server systems and batch process systems. ERP systems keep evolving continuously in terms of technology used and functionality offered. Migrating between new and old versions of an ERP package is problematic when the versions are not backward compatible, or when an organization has made modifications to the earlier installed version.

Thus, an ERP package has many detailed options, parameters, capabilities and models built into it when developed by a vendor. An organization implementing an ERP package must clearly understand, identify, and outline its objectives in this implementation, and its requirements and expectations from the ERP package. It is important that the organization match these expectations against the solutions provided by different ERP packages, and select the appropriate ERP vendor whose approach aligns best with the organization's underlying business model, business practices, and requirements. Some software vendors providing ERP packages today are listed in Table 2.2.

SAP	http://www.sap.com
Computer Associates International	http://www.cai.com
Systems Software Associates	http://www.ssax.com
Oracle	http://www.oracle.com
Baan	http://www.baan.com
Peoplesoft	http://www.peoplesoft.co.nz
JD Edwards	http://www.jdedwards.com

Table 2.2 ERP Software

For example, SAP R/3 is a commonly used ERP application in many industries. It consists of many modules that link operational steps [8]. They can be used alone or combined with other solutions. Some of the modules that SAP R/3 has are listed in Table 2.3.

By implementing ERP applications, organizations aimed to replace complex, disparate, obsolescent systems, improve competitive performance, and improve the poor quality and visibility of information. ERP applications help organizations track customers, money, materials, assets, labor, utilization, etc.

ERP systems have their shortcomings. They are built for recording events that have already occurred, rather than planning for what will be. Thus, they are good at record keeping but not at intelligent decision-making. They can accommodate complex workflows, but lack the ability to adapt and restructure with changes in surroundings. They also lack the ability to scale to large volumes since their order taking capacities are limited. While they integrate multiple business functions, they lack the ability to expand their scope to multiple enterprises. Lastly, tracking of enterprise information is possible on the basis of data entered into ERP databases in different units of the enterprise. The solution provided by an ERP application is accurate to the extent data in the ERP database is accurate. Thus, it is necessary that for optimal decision-making, data in the records be accurate and real time. We discuss methods of data acquisition and records keeping later in this chapter.

Through the 1990s, most large industrial companies have installed ERP systems. ERP systems manage and share data within an enterprise – they manage the "internal supply chain". Companies today need to plan across a wider span of activities and make trade-offs to optimize the "overall" profitability. This requires sophisticated systems that analyze the interplay of complex interactions across enterprises – between the company, its suppliers, its distributors, and other trading partners in its "supply chain". In the next section, we analyze supply chains and the changing objectives of a company as a trading partner in today's supply chains.

Financial Accounting	Collect data relevant to accounting, providing complete documentation and comprehensive information, and an up-to-the-minute basis for enterprise-wide control and planning.
Treasury	Module for financial management aimed to ensure liquidity of the company worldwide, structures financial assets profitably, and minimizes risks.
Controlling	An array of compatible planning and control instruments for company-wide controlling systems, with a uniform reporting system for coordinating the contents and procedures of company's internal processes.
Enterprise Controlling	Continuously monitors company's success factors and performance indicators on the basis of specially prepared management information.
Investment Management	Offers integrated management and processing of investment measures and projects from planning to settlement, including pre-investment analysis and depreciation simulation.
Production Planning	Provides comprehensive processes for all types of manufacturing: from repetitive, make-to-order, and assemble-to-order production, through process, lot and make-to-stock manufacturing, with functions for extended MRP II and electronic kanban, plus optional interfaces to process control systems, CAD, etc.
Materials Management	Optimizes all purchasing processes with workflow-driven processing functions, enables automated supplier evaluation, lowers procurement and warehousing costs with accurate inventory and warehouse management, and integrates invoice verification.
Plant Maintenance and Service Management	Provides planning, control, and processing of scheduled maintenance, inspection, damage-related maintenance, and service management to ensure availability of operational systems, including plants and equipment delivered to customers.
Quality Management	Monitors, captures, and manages all processes relevant to quality assurance in the entire enterprise, coordinates inspection processing, initiates corrective measures, and integrates laboratory information systems.
Project System	Coordinates and controls all phases of a project, in direct cooperation with Purchasing and Controlling, from quotation to design and approval, to resource management and cost settlement.
Sales and Distribution	Actively supports sales and distribution activities with outstanding functions for pricing, prompt order processing, and on-time delivery, interactive multilevel variant configuration, and a direct interface to Profitability Analysis and Production.
Human Resources Management	Provides solutions for planning and managing your company's human resources, using integrated applications that cover all personnel management tasks and help simplify and speed the processes.
SAP Business Information Warehouse	This independent data warehouse solution summarizes data from R/3 applications and external sources to provide executive information for supporting decision-making and planning. Reports covering a wide range of information requirements, automated data staging, and standard R/3 business process models ensure rapid implementation and low operating costs. This "return on information" means that the SAP Business Information Warehouse soon pays for itself.

Table 2.3 SAP R/3 Modules [8]

2.5 Supply chain management

In the early 1990s, the phrase "supply chain management (SCM)" came into use. The original motive of SCM was "elimination of barriers between trading partners" in order to facilitate synchronization of information between them. Recent literature offers many variations on the same theme when defining a supply chain. A working definition

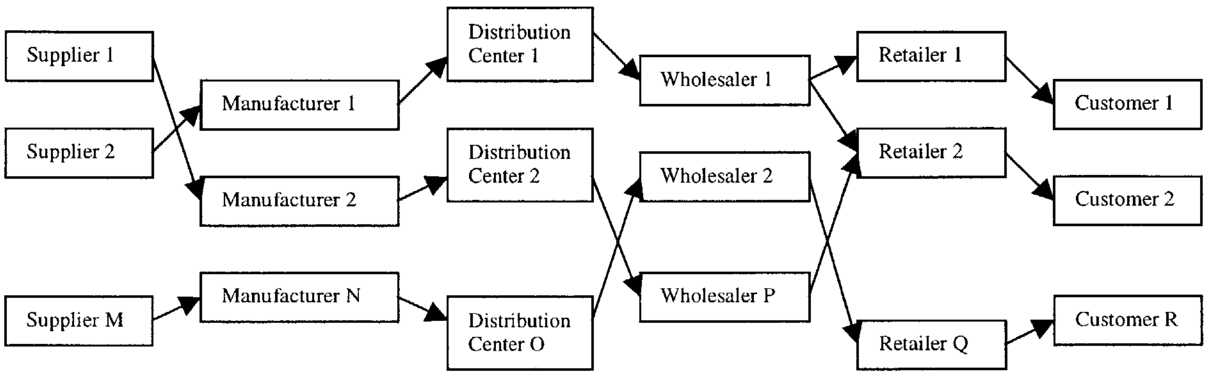


Figure 2.2 A simple supply chain.

by Stevens [9] defines a supply chain as "a connected series of activities that is concerned with planning, coordinating and controlling materials, parts, and finished goods from supplier to customer. It is concerned with two distinct flows (material and information) through the organization." Many authors consider strategic decision-making and systems integration a differentiating virtue of supply chains. Still others consider carriers, and sometimes even the government, as integral components of a supply chain. Tayur, *et al.*. [10], provide a comprehensive review of supply chain literature, including definitions of the terms used in this field.

Figure 2.2 shows the structure of a typical supply chain. It consists of a number of units – beginning with suppliers, who provide raw materials to factories or manufacturing plants, which manufacture products and send them to distribution centers. These transport them to regional distributors or wholesale dealers, who ship them to retailers. The end of a traditional supply chain is usually the customer, who buys products from the retailer. Although this composition is typical, supply chains vary in length. Different industries might have slightly differing structures of their supply chain. A manufacturing giant might have a highly structured distribution network – comprising of central warehouse, regional warehouses and local warehouses – through which a product goes before it reaches the retailer. Or, a small, regional company may suffice with having just one distributor for supplying products to its retailers. An entire supply chain could exist within one company. Or a supply chain can span multiple enterprises before it reaches the customer. Traditionally, planning, purchasing, manufacturing, distribution, and marketing operated independently along the supply chain. Each activity had its own set of objectives and often, these objectives were conflicting (for example – manufacturing operations may be designed to maximize throughput and minimize costs, with little consideration for inventory levels, distribution capabilities or market demand). Supply chain management has evolved as a strategy to coordinate activities of these independent functions, and create a single, integrated plan for the entire organization.

Figure 2.3 Supply chain decisions

In a supply chain, decisions taken are usually classified as strategic, tactical, or operational. Strategic decisions are usually linked with the company's corporate strategy, and guide the design of the supply chain. They are typically made over a long period of time (2-5 years or more), and traditionally involve all partners in the supply chain. Tactical decisions are taken on a monthly to annual basis, whereas operational decisions are short term, and directly affect day-to-day activities. Tactical and operational decisions are traditionally taken independently in individual units of the supply chain – in a warehouse, on the shop floor in a factory, etc. They deal with forecasting, procurement, production and inventory management, warehousing and distribution, and logistics issues. For example, in a soap factory, deciding which type and what quantity of soaps should be produced during the current week, and which machines and assembly lines should be used for this purpose, are operational decisions. MRP, MRP II and ERP help enterprises take tactical and operational decisions.

Every company performs five basic activities in its supply chain: buy, make, move, store and sell. There are a multitude of decisions, strategic, tactical and operational, to be taken at each of these five actions. These are listed in Table 2.4. Supply chain simulation tools are commonly used to derive the optimal answers to strategic issues proposed in Table 2.4. The most common representation of a supply chain is a multi-commodity, multiple sources, and multiple sinks network. Hicks [11] proposes a four step methodology for using simulation and optimization techniques for reaching these strategic decisions. The network is first modeled as a set of nodes and arcs, and a linear-mixed-integer-programming problem is formulated. Powerful optimization solvers are applied on this network to evaluate total costs. Step 2 consists of simulating the network over a period of time in order to observe its behavior. The result is a supply chain network design, including the structure and a proposed policy scheme. In step 3, the policy is optimized using simulation-optimization. The final step is testing the design for robustness.

Activity	Strategic Decision	Tactical & Operational Decision
Buy	Choosing suppliers, long term contracts vs. short term deals	Type and quantity of raw material to be purchased, date, time and location of arrival
Make	Factory locations, Product lines, Proximity to end customer	Scheduling production, allocating resources
Move	Setting up transportation network, outsourcing vs. in-house function	Planning optimal routes for trucks
Store	Distribution network design, warehouse locations	Loading / unloading operations, book-keeping
Sell	Demand forecasting, special promotions	Order fulfillment, customer service

Table 2.4 Supply chain activities.

The objective of supply chain management activities is to meet customer demand for guaranteed delivery of high quality, low cost, customized products with minimal lead-time. The attempt is to improve responsiveness, understand consumer demand, intelligently control the manufacturing process, and align together the objectives of all partners in the supply chain. To achieve this objective, companies need to have visibility into the entire supply chain of transaction and planning systems – of its own plans as well as those of its suppliers and customers. Also, the company should be flexible enough that it can adjust, rebuild and re-optimize plans in real time, to take care of unexpected events taking place in the supply chain.

These needs have propelled the development of optimization packages for managing the entire business process, beginning with advanced planning and scheduling (APS) tools to help companies match their supply with demand, and later integration with modules for customer relationship management and product lifecycle management. We begin with a review of APS tools in the next section.

2.6 Advanced Planning and Scheduling (APS)

Supply chain management requires responsive, intelligent decision support tools to determine optimal (or at least feasible) methods of satisfying customer demand and product supply. APS tools have been developed with the intention of meeting this requirement. These systems aim at optimizing the supply chain objective discussed in the previous paragraph subject to constraints of resource availability, capacity costs, labor and materials costs, and transportation resources. They help companies forecast demand with the help of sophisticated modeling and statistical techniques [12]. Given their memory-resident and exception based nature, together with their object-oriented design and algorithmic domain expertise, they are technologically far superior to ERP and give results very fast compared to ERP. APS tools are designed to help companies create plans and schedules that are based on system constraints. APS tools generate a high rate of return by shortening the forecast cycles, enhancing visibility of production plans and schedules, increasing accuracy of order date commitments, taking real-time decisions in the face of supply / demand fluctuations, and planning in real-time rather than batch processing.

APS fails to yield the above-mentioned improvements if companies do not adopt new procedures and modify their business processes simultaneously. Although APS tools are "intelligent", they are hampered due to their focus on manufacturing, distribution and transportation functions in the supply chain. This focus is acceptable in traditional, slow moving environments. The current fast-paced business environments warrant a broader scope of planning efforts. Where customization and perfect delivery are the price for getting business, customer interaction is the main driver behind the entire product delivery process [13]. Software vendors have moved on from pure APS tools to building a complete set of software for business process optimization.

2.7 Business process optimization (BPO)

Companies like i2 Technologies [14], planning and optimization software vendors, have developed software solutions for business process optimization (BPO). BPO is a class of decision-intelligence software that features multi-enterprise optimization and integration. ERP, legacy and other transaction systems, are built for recording what already happened. BPO software leverages current infrastructure, by deriving raw data from ERP systems or any other existing data source. Next, it engages an integrated set of planning engines to produce an optimal solution based on a complete view of the enterprise and its trading partners. Last, it feeds the optimal solution data back into the transaction system for execution. Its major components typically include the following [14]:

- Product life cycle management: this spans the entire product development and product lifecycle process – from early concept definition, through development and test, launch, to product phase-out. It provides support for strategic issues and daily operations.
- Supply chain management, this includes:
    - Demand fulfillment – to provide fast, accurate, and reliable responses to customer orders. It is mainly an execution-level process that includes order capturing, customer verification, order promising, backlog management, and order fulfillment.
    - Demand planning – to understand customers' buying patterns and develop aggregate, collaborative forecasts. It is by definition a planning process that feeds into the supply planning process, and subsequently the demand fulfillment process. It involves long-term, intermediate-term and short-term time horizons.

    - Supply planning – to optimally allocate enterprise resources to meet demand. This is a planning-level process that spans the strategic and tactical supply-planning processes. It includes long-term planning, inventory planning, distribution planning, collaborative procurement, transportation planning and supply allocation.
- Customer Relationship Management (CRM):
    - Creating demand through identifying and acquiring customers, and developing marketing content and offers.
    - Matching demand with customized product offerings.
    - Fulfilling demand by executing the sales transaction (either directly, or through indirect channels), and providing real-time, integrated order fulfillment.
    - Managing long-term customer relationships, by servicing customer needs and cross-selling and up-selling opportunities.
- Inter-process planning includes:
    - Integrated Sales and Operations Planning – ability to review the operation plan with the revenue objectives for the financial periods – based on the different plans of the different authority domains – including promotion plans, new product introduction plans, possible long-term contracts etc.
    - Financial Planning – ability to project revenues, earnings and other financial measures for the next few financial periods based on the plans of the different authority domains with the organization on a continual basis and changes in the market conditions. It will also be able to suggest corrective actions to alleviate the deviations from the strategic plan. This will help in monitoring metrics for different authority domains of the organization to provide them a quick feedback on their impact on the entire financial plan.
- Strategic Planning: enables companies to plan scenarios, set goals, and monitor the performance.

Here, it is necessary to note that BPO software gets data from traditional legacy systems or ERP software. Hence, problems of data management associated with ERP and legacy systems apply directly to BPO software too. These data management issues are discussed later in this chapter. In the next section, we review the concept of CPFR.

2.8 Collaborative planning, forecasting and replenishment (CPFR)

MRP, MRP II, ERP, and to some extent, APS helped companies to achieve efficient production and operations planning and scheduling. We observe an increasing trend towards improving demand forecasting and fulfillment through the development of APS to BPO to CPFR. CPFR attempts to determine the right number of specific products to put in an individual store on a particular day of the year [15]. Unpredictable factors such as weather, transportation delays, production problems, and administrative errors can all wreck havoc on supply and demand. Product promotions create massive swings in demand. Suppliers are forced to carry unprecedented amounts of safety stock, or stay lean and risk being unable to fulfill demand. The first option raises costs for everyone; the second results in lost sales, and frustrates customers. Also, reducing inventory costs returns hidden savings in the supply chain, due to factors like freeing up of fixed assets, capital costs for inventory, manufacturing inefficiencies, redundant handling and transportation, and improved customer service.

CPFR takes an integrated approach to supply chain management among a network of trading partners. Trading partners share forecast and results data. CPFR software analyzes this data, and alerts planners at each company to exceptional situations that could affect delivery or sales performance. Trading partners then collaborate to resolve these exceptions by adjusting plans, expediting orders, correcting data entry errors. Representatives from retail stores, manufacturers and consulting firms have come together to form a Voluntary Inter-industry Standards Committee (VICS) to take the leadership role in improving the flow of product and information through the entire supply chain [16]. The VICS Committee volunteers detailed guidelines for companies to implement CPFR. It also provides a CPFR roadmap for companies to supplement the voluntary guidelines. These are described in detail in [16].

If implemented correctly, CPFR improves data communication among trading partners. It improves forecasting and planning by providing the ability to see planned results. A retailer can prevent out-of-stock situations, especially during product promotions. A manufacturer can optimize product mix, promotion timing, and margins across supply chains. Better partnership with suppliers results in lower inventory levels at retailers. Manufacturers can make-to-demand rather than make-to-stock, and offer savings in inventory and production costs, and product obsolescence costs.

Implementation of CPFR begins with an understanding amongst trading partners to develop specific plans in different product categories based on best practices. It is essential that both partners own the agreed upon plans and processes in order to achieve success. The jointly accepted plan determines which product should be sold when, where and how. To do this, the plan is implemented in each company using existing planning and scheduling systems, and is required to be made accessible to either party.

Similar to all software systems discussed before, the successful implementation of CPFR is dependent on data available to existing systems at each trading partner, and their ability to communicate with each other. This ability is dependent on data representation standards and modes of data communication used by involved parties. Usually, each system has its unique method of representing data making it difficult to be used directly by another system. Therein stems the need for an industry standard for data representation.

2.9 Data acquisition techniques

As discussed earlier, the success of all software applications in terms of delivering what they promised depends on the availability of the right data at the right place at the right time, in the right format. Data inputs are usually required from the supply chain unit under consideration (internal data) and sometimes from units other than the one under consideration (external data).

Internal data inputs include transactions associated with receiving material, staging, storage, location, picking, manufacturing, product status, etc. These transactions are processed by personnel in goods receiving, material handling, storage, accounting, inspection, shipping, and order entry [17]. Material planners, schedulers, order entry and manpower planning personnel process conversion transactions. Identification transactions include item identification, specifications, order numbers, locations, update transactions, etc. These transactions may influence the status, schedule, or location of the product.

External data transactions involve purchase, acquisition, movement and location of items in the supply chain – including purchase order data, advance-shipping notices, etc. Buyers, suppliers, transporters and receiving personnel, all of who are responsible for data integrity, process these transactions.

In today's manufacturing setups, the time span between an event taking place and its registration and availability in the computer system is unacceptably long. The lag from the time an event occurs until the time it gets registered into the system results in proliferation of bad (un-updated) data and the creation of a blank spot in information visibility. Companies cover themselves by using safety stocks and safety lead times. As a result, inventory increases. A complementary problem is the entry of faulty information, *i.e.*, the data captured can contain faulty amounts (ex: 900 instead of 800) or wrong part numbers (ex: B2413 instead of 82413). This gives rise to poor record accuracy. Data accuracy is vital for any company planning its production using enterprise systems [18]. For example, an inventory record which reports inventories lower than actual can trigger an unnecessary order that drives up inventory and wastes capacity. A record higher than actual can result in stock-outs and perhaps work stoppages. Also, for optimal utilization of the unit's resources, it is essential that accurate information on resource status be maintained.

Traditionally, shop floor operators enter data into workbooks or sheets of paper, after which data-entry operators type it into a computer. Alternatively, shop-floor operators type in data into computer terminals located on the shop floor. Studies show that with this technique, there is likelihood of 1 error for every 300 characters entered [18]. Another venue for errors is the transcribing of data during entry. Also, real-time data acquisition is not achieved because of the inherent delays in this method – between the gathering and entry of data.

To overcome these disadvantages, automatic data capture techniques have been developed, and are under development. These techniques make it possible to enter a stream of data using automated operations. This can be achieved by multiple means – for example, by expressing the data in a machine-readable code, information can be entered in a computer with a code-reading device; or by using voice data entry system, where a voice recognition agent captures data from the words spoken by the operator; by machine vision, other optical systems, or mechanical and inductive flags. A software application in the computer can then process the information further. Data is transferred from point to point using manual methods, electronic data interchange (EDI), virtual private networks (VPN) or the internet. We discuss these briefly below.

2.9.1 Barcodes

The most common example of automating data acquisition is the use of barcodes. A barcode symbol consists of a series of parallel, adjacent bars and spaces. Predetermined width patterns are used to code actual data into the symbol. To read information contained in this symbol, a scanning device is moved across the symbol from one side to the other, or triggered with a button. As it is moved or triggered, the barcode decoder analyzes the barcode width patterns of bars and spaces, and the original data is recovered. Typically, laser scanners can read barcodes from near contact till 12 inches distance (very powerful scanners go up to as much as 4 feet).

The most popular barcode format is the Universal Product Code (UPC) Format, which is found on almost all products today. Developed by the Uniform Code Council (UCC), and available since the early 1970s, this format is universally recognized [19]. Barcodes offer speed, accuracy, efficiency, and consistency in data acquisition. A study conducted by Swamidass [20] presents ample industry evidence about the use and benefits of barcodes as an enabling technology in manufacturing environments, and its contribution to manufacturing cost reduction, overall quality improvement, cycle-time reduction, and improved profitability.

Although the barcode offers a lot of benefits, it has its drawbacks – it stores limited amount of data, it needs to be maintained clean so that the reader can read the bars and spaces, the object should be oriented properly such that a line of sight should be established between the barcode and the reader, they cannot be embedded into products or pallets, operator intervention is required to read a barcode, barcodes can be easily copied, allowing counterfeit use and compromise, etc. Radio frequency identification tags, also known as RFID tags, are widely emerging as a complementary technology to help overcome these drawbacks.

2.9.2 Radio frequency identification (RFID) tags

RFID tags basically consist of a transponder that is electronically programmed with unique data. Data is read/written on the tag through an antenna or a coil by a transceiver (with a decoder), which is connected to a host computer. RFID tags and antennas come in a variety of sizes and shapes. The tags are categorized as active or passive. Active RFID tags are powered by an internal battery and are typically read/write and have various memory sizes. The battery-operated power gives the tag a longer read range, with a trade-off of greater size, greater cost and limited operational life. Passive RFID tags operate without a separate external power source and obtain operating power generated from the reader. As a result, they are lighter, cheaper and long lasting as compared to active tags. They usually have short read range, and require a high-powered reader. They are typically read only, and are programmed with a unique set of data (usually 32 or 128 bits) [21].

The significant advantage of RFID tags is the non-contact, non-line-of-sight nature of the technology. Tags can be read through a variety of substances like snow, fog, paint, and other visually and environmentally challenging conditions, where barcodes and other technologies are useless. They can also be read in challenging circumstances at remarkable speeds (less than 100 ms). They possess read/write capability. No operator intervention is necessary. Tags have an anti-collision feature, which allows for reading of multiple tags at once. They can be read from significant distances, too. Most importantly, it is not possible to copy RFID tag data through mechanical means, and by using encryption techniques, unauthorized replication will be virtually impossible.

Today, companies like Texas Instruments (TI) and Philips Semiconductors are developing and producing RFID technology. Tag costs (active tags - $5-50, passive tags - $2-3 [22]), compatibility, ease of use, and open standards, are important concerns in bringing this technology into everyday use. TI plans to get tag costs down to 50 cents using proprietary Tag-it technology and production volumes in the order of millions.

Motorola is developing "Bistatix" technology with the objective of reducing the RFID tag prices and making them affordable [23]. Bistatix works on a capacitive coupling principle, where-in, electric fields are capacitively coupled to and from a reader and tag. It consists primarily of an IC chip connecting antennae printed on a sheet of paper. As in an inductive system, a Bistatix reader/writer generates an excitation field, which serves as both the tag's source of power and its master clock. The tag cyclically modulates its data contents and transmits them to the reader's receiver circuit. The reader demodulates and decodes the data signal and provides a formatted data packet to a host computer for further data processing. Because of its simple design and construction, these tags are simple to manufacture. This reduces production cost. The tags also offer stability of operation due to absence of a capacitor, and offer robustness as long as the IC remains intact. The tag can be bent, cut, torn; as long as some remnants of the electrode are connected to the silicon chip. In terms of tag orientation, they perform optimally at parallel planes of reader and tag, but monopole-coupled bistatix tags do not require this orientation constraint. They can be applied to any physical configuration, are very flexible, thin, flat, and not limited with regard to the substrate material.

Thus, Bistatix tags eliminate some of the drawbacks of inductive RFID tags – they are low-cost and easy to make and use. The issue of compatibility and standards remains to be resolved.

2.9.3 Electronic data interchange (EDI)

EDI is a member of a larger family of technologies used for communicating business messages electronically, including facsimile, electronic mail, telex, and computer bulletin boards. EDI is commonly defined as application to application transfer of business transactions on a computer.

EDI implementation involves understanding EDI standards, communications link between partners, and available software. EDI standards developed beginning in 1960, when proprietary standards were implemented and organizations were created to develop industry and inter-industry standards. Use of EDI increased dramatically during late 1970s and early 1980s, and ANSI ASC X12 (American National Standards Institute, Accredited Standards Committee) was chartered to develop standards to facilitate electronic interchange of business transactions. EDI implementation involves a communication medium for electronically transferring data between organizations. There are many popular channels for EDI communication. At the top of the list are Value Added Networks (VANs), which are similar to electronic mailboxes, as they provide postbox service between EDI users (ref 2.21). VANs have the capability to exchange data with a wide variety of computers using appropriate communication protocols. VANs are popular with small sized companies, running low volumes of transactions. Companies running high volumes of transactions prefer Direct Connections, which are usually telephone lines connected with modems at both ends, dedicated solely for the purpose of transactions transfers. EDI software is the front-end for interaction with people. The software package should allow data to be entered and encoded into an EDI standard format, and also decode incoming EDI data and convert it to in-house data formats.

Common use of EDI is in sales, order processing and purchasing, inventory management, distribution, financial management, etc. Implementation aspects have more to do with managerial support than technical implementation. Trading partner agreements, vendor agreements, VAN agreements, role of lawyers and auditors, and security of communication are some of the issues of concern. Costs associated with EDI include hardware costs (computers, VAN, and appropriate software) depending on scale of implementation. These costs are quite large (average hub investment of $1 million, plus spokes investment of an average of $45,000 [24]). This is coupled with adjustment time, and lack of human resources skilled in using EDI. Using EDI requires a company to also educate its trading partners to use it. Company data structures sometimes do not fit standard EDI form, which forces manual intervention in the process [25]. Integration of legacy systems poses a big problem before companies using EDI. Also, EDI standards are inflexible. VANs are costly too. Also, each transaction in EDI is in a separate format, causing VAN costs to rise higher. Large companies have annual EDI transactions to the order of 100 million, VANs charge companies on a transactions basis. Hence companies are looking for the cheapest network to conduct their transactions.

Although EDI is still better than paper based transactions, it doesn't lend itself to change. Today's business models emphasize on speed of transactions, reducing product lifecycle, having multiple partners in the supply chain, and a strong collaborative focus. EDI is traditionally a hub and spoke architecture (with VANs), emphasizing growth with trading partner, slow to change standards, has limited capabilities, and requires experts to implement it. The internet offers a very cost-efficient replacement for VANs used for EDI. As a result, companies are gradually moving towards using EDI together with the internet to cut costs, and later eliminate use of EDI totally and move on to internet based business-to-business services for conducting all their transactions. Companies are also steadily migrating towards using extensible Markup Language (XML) for representing their data. Still, issues like standardization of internet based transaction protocols, security and encryption of data transfer on the internet, representation of complex data, etc. are points of concern which we shall discuss in more detail in Chapter 4. In the next chapter, we study the dynamics at play in a typical manufacturing supply chain, under varying conditions of information visibility and trading partner collaboration.

Chapter 3

Modeling supply chain dynamics

3.1 Introduction

As seen in the previous chapter, one of the fundamental requirements for optimizing supply chain performance is availability of the right data at the right place and the right time in the right format, *i.e.*, visibility of appropriate information in the supply chain. This, coupled with increased willingness for collaboration between trading partners has been observed to drive down costs in the supply chain. In this chapter, we investigate these claims by modeling a supply chain for a beer manufacturer – what is popularly known as the beer game. The earliest description of the game dates back to the work of Forrester [26] in industrial dynamics. The System Dynamics group at MIT further developed the game to the format it is played in today. Many researchers in supply chain management and system dynamics have studied the beer game. Logistics executives at Proctor and Gamble, while examining the order patterns of their product – pampers, observed an increase in the amplitude of fluctuation in the orders placed at the retailers to the orders placed at the factory. They called this phenomenon the bullwhip effect [34]. The bullwhip effect is a repeatable observation in many supply chains that lack information visibility, including the beer game. I played the beer game as part of a Logistics Systems class exercise at MIT, and had a first hand experience of supply chain dynamics in action. In this chapter, we simulate the supply chain for a beer manufacturer under different conditions of information visibility and collaboration with the help of system dynamics models. We present our conclusions at the end of this chapter.

3.2 The beer game

The "Beer Game" is a realistic simplification of the supply chain for beer manufacture. The game is popularly used as an introduction to systems thinking, dynamics, computer simulation, and supply chain management. It has been played by thousands of people, all over the world from high-school students to CEOs of major corporations [27]. It serves as an excellent experiment for studying the effect of system microstructure (individual behavior and decision-making under given circumstances) on supply chain dynamics [28].

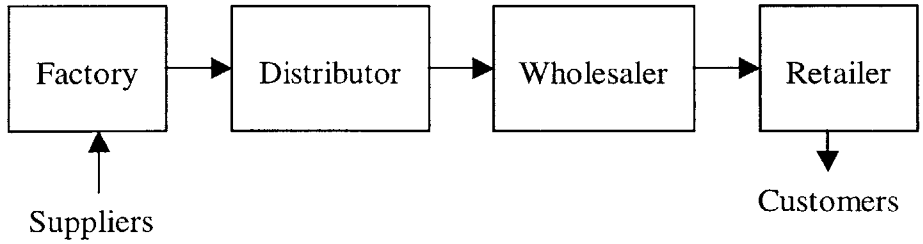


Figure 3.1 The beer distribution network.

The beer distribution network under consideration consists of supplier, factory, distributor, wholesaler, retailer and end customers. It is shown in Figure 3.1. Each week, customers place demand with the retailer, who fulfils it from his inventory. He, in turn, orders beer from the wholesaler, who provides it out of his current inventory. The wholesaler gets his supply from the distributor, who gets his supply from the factory. The factory brews beer. Every week, inventory in each unit keeps depleting, and the participants replenish the inventory to maintain required safety stock levels. There are ordering, shipping, production and communication delays between each of these units in the supply chain. Each unit incurs costs for maintaining inventory and running into backlogs. The objective is to minimize supply chain costs by trying to maintain inventory at safety stock levels while meeting incoming demand.

Traditionally, the beer game is played on a board representing the beer distribution network. A sample board is shown in Figure 3.2. There are four individual units: Factory (F), Distributor (D), Wholesaler (W), and Retailer (R).

Each box between units in Figure 3.2 represents one week of delay – thus, there is a two-week delay between an order being shipped by the distributor and the wholesaler receiving it. Similarly, there is a two-week delay between an order placed by the retailer

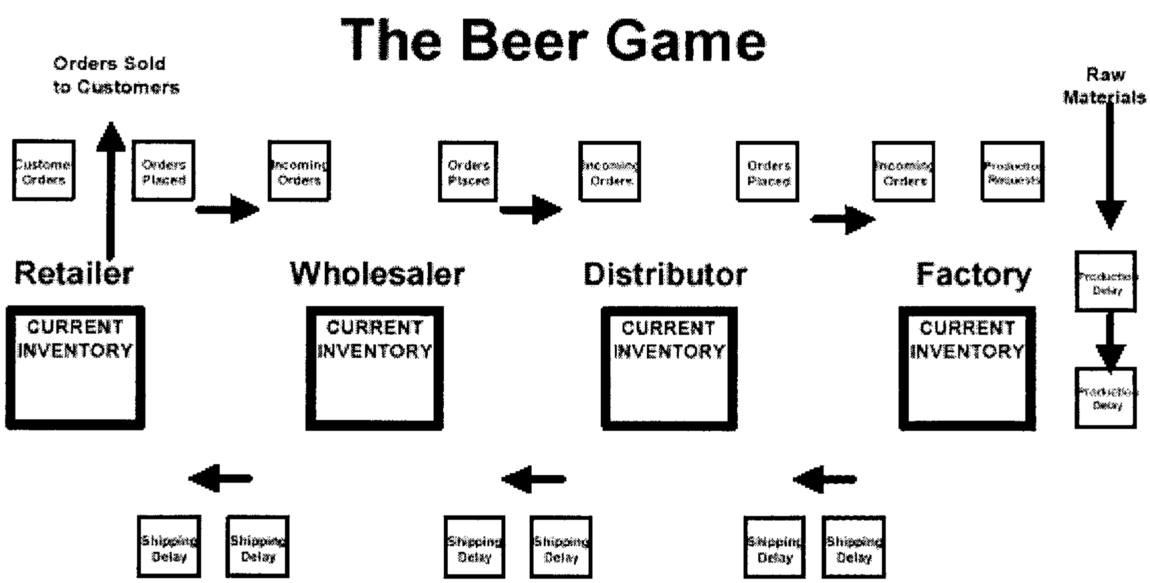


Figure 3.2 The traditional beer game board. [30a]

and the wholesaler being notified about it. Each unit in the supply chain acts independently, without collaboration, as is common practice in traditional businesses, and hence a given unit does not have much information about inventories held at other units in the supply chain at any given time. The only interaction between units is receiving orders and shipping crates to customer, and placing orders and receiving crates from the supplier. The supply chain simulated above is far simpler than a real life supply chain. Some of the simplifications include:

- It consists of only four independent units.
- We exclude random events – for ex: machine failures, transportation issues, strikes, etc.
- We do not impose capacity constraints
- We do not impose financial constraints, etc.

Detailed instructions on how the game is played can be found in [28], [29] or [30]. In the next section, we explain the dynamics in the supply chain for beer with the help of a Vensim model.

3.3 A system dynamics model for the beer distribution network

The beer game described above represents the supply chain in a typical manufacturing industry. It is a good example of a system with simple structure, low combinatorial complexity, and high dynamic complexity – arising due to interaction amongst agents over time [31]. Being a dynamic system, the individual variables are tightly coupled together, and interact strongly with one another and the rest of the system. They are strongly governed by feedback. Multiple factors influence their values, introducing non-linearity into the system. The future values of many variables depend on their historical values. In the traditional game, there are multiple points where delay is introduced into the system. Delays create instability and oscillations in the system, causing inventory levels to significantly sway from the expected safety stock level. (Adding delays makes system eigenvalues complex conjugates, yielding oscillatory solutions, and relevant system parameters determine whether the oscillations are damped or exponential [31].)

In this section, we describe the model used for simulating the beer game. Kirkwood [33] has built a Vensim model for the beer distribution network based on Sterman's description of the beer game as designed at MIT [28]. The model is shown in Figure 3.3.

Each unit in the beer game has a supplier (except for the raw materials suppliers) and a customer (except for the end customer). Since there isn't any collaboration between individual units in a traditional beer game, each unit observes the demand patterns of its customer and places orders accordingly with its supplier.

The game begins at time t = 0. At every incremental week, each unit makes a decision about how much it will order, and how much it can ship in that time period. The forecasting technique used for ordering beer crates from other units in the supply chain is as described by Sterman [28]. The decision rule utilizes information locally available to the decision maker.

At each unit, the Effective Inventory, EI, at time t, is given by

$$EI(t) = I(t) - B(t) \qquad (3.1)$$

$I(t)$ = Inventory, given by

$$I(t) = {}_0\int^t (\Delta inv(\tau)/\Delta\tau)\bullet d\tau$$

$$\Delta inv(\tau)/\Delta\tau = i(\tau) - s(\tau)$$

Integrated numerically:

$$I(t) = I(t\text{-}1) + i(t) - s(t) \qquad (3.2)$$

$\Delta inv(\tau)/\Delta(\tau)$ = change in inventory during each time interval
$i(t)$ = incoming orders
= crates sold by the supplier, received with delay
$s(t)$ = crates sold by this unit
= minimum required to satisfy customer orders
note that $I(0) = 12$

$B(t)$ = cumulative backlog at the unit, is given by

$$B(t) = {}_0\int^t (\Delta b(\tau)/\Delta\tau)\bullet d\tau$$

$$\Delta b(\tau)/\Delta\tau = o(\tau) - s(\tau)$$

Integrated numerically:

$$B(t) = B(t\text{-}1) + o(t) - s(t) \qquad (3.3)$$

$\Delta b(\tau)/\Delta\tau$ = backflow at each unit, i.e., the accumulation of backlog during in each time period
$o(t)$ = orders placed at this unit
note that $B(0) = 0$

In general, Orders placed by a unit, O, at time t, are given by

$$O(t) = max\,(0, IO(t)) \qquad (3.4)$$

$IO(t)$ = indicated order rate, is computed based on three factors:

- expected losses from the stock (L)
- discrepancy between desired and actual stock (AS)
- discrepancy between desired and actual supply line (ASL)

$$IO(t) = L(t) + AS(t) + ASL(t) \qquad (3.5)$$

$$L(t) = x_l \bullet L_{t-1} + (1\text{-}x_l) \bullet L_0 \qquad (3.6)$$

$$AS(t) = x_s \bullet (S^* - S(t)) \qquad (3.7)$$

$$ASL(t) = x_{sl} \bullet (SL^* - SL(t)) \qquad (3.8)$$

x_l, x_s, x_{sl} = weight factors determined through regressive expectation
$S(t)$ = stock of crates
S^* = desired stock [=12 in beer game]
$SL(t)$ = supply line
SL^* = desired supply line
= expected lag • desired throughput

Applied to the beer game, the equation for Orders placed, $O(t)$, is given by

$O(t) \quad = max\ (0, IO(t))$

$IO(t) \quad =$ Demand Forecast + actual stock gap + actual supply line gap

$IO(t) \quad =$ Demand Forecast + $A \bullet (12 - EI(t)) - B \bullet Spl(t)$ (3.9)

$Spl(t) \quad = {}_0\int^t (\Delta sf(\tau)/\Delta\tau).d\tau$

$\Delta sf(\tau)/\Delta\tau \quad = O(\tau) - i(\tau)$

Integrated numerically:

$$Spl(t) = Spl(t\text{-}1) + O(t) - i(t) \qquad (3.10)$$

$Spl(t)$ = supply line gap

$\Delta sf(\tau)/\Delta\tau$ = accumulation in the supply line during every time interval

A, B = weight factors

The Demand Forecast function would typically be either the SMOOTH or FORECAST function of Vensim. The time delays for the crates to move between units are accounted by the FIXED DELAY function. The equations described above are representative of one unit in the supply chain. The complete set of equations describing the model drawn in Figure 3.3 is listed in Table 3.1. Solving them becomes quite complex due to the interdependence of many variables in the model.

We ran the simulation for a span of 36 weeks, the typical time period for which the beer game is played. Figure 3.4 shows the effective inventory levels at all units in the supply chain over the simulated period. Figure 3.5 shows the orders placed by individual units during this time. Inventory costs are assumed to be 0.50 $/unit/week and backlog costs are assumed to be 1.00 $/unit/week [28]. The customer demand was kept constant at 4 crates of beer for the first 4 weeks. A step increase to 8 crates was applied at the 5th week, and the demand was maintained constant at 8 throughout the rest of the simulation.

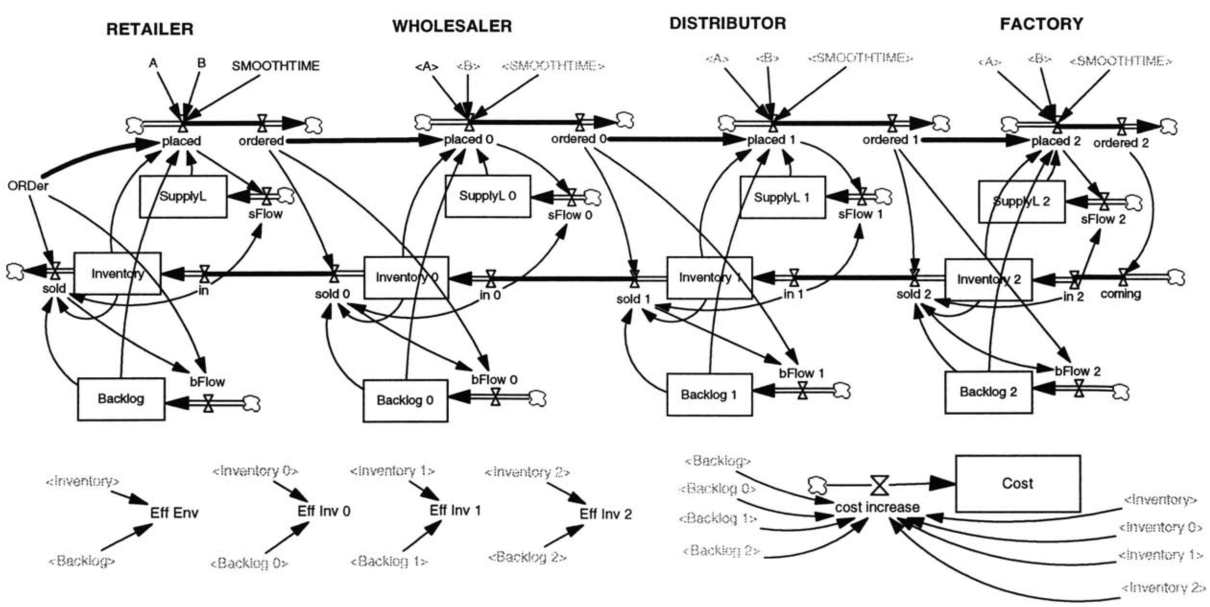


Figure 3.3 Vensim model for the traditional beer game.

(Developed by Prof. C. Kirkwood at ASU. Based on Sterman's description of the beer game developed at MIT.)

No:	Variable and defining equation	Comments
(01)	A = 0.25	Forecasting parameter
(02)	B = 0.33	Forecasting parameter
(03)	Backlog = INTEG(bFlow , 0)	Backlog at retailer
(04)	Backlog 0 = INTEG(bFlow 0 , 0)	Backlog at wholesaler
(05)	Backlog 1 = INTEG(bFlow 1 , 0)	Backlog at distributor
(06)	Backlog 2 = INTEG(bFlow 2 , 0)	Backlog at factory
(07)	bFlow = ORDer - sold	Accumulation of backlog at retailer
(08)	bFlow 0 = ordered - sold 0	Accumulation of backlog at wholesaler
(09)	bFlow 1 = ordered 0 - sold 1	Accumulation of backlog at distributor
(10)	bFlow 2 = ordered 1 - sold 2	Accumulation of backlog at factory
(11)	coming = ordered 2	Materials in transit to factory
(12)	Cost = INTEG(cost increase , 0)	Total supply chain cost
(13)	cost increase = 1 * (Backlog + Backlog 0 + Backlog 1 + Backlog 2) + 0.5 * (Inventory + Inventory 0 + Inventory 1 + Inventory 2)	Weekly supply chain cost
(14)	Eff Env = Inventory - Backlog	Effective Inventory at retailer
(15)	Eff Inv 0 = Inventory 0 - Backlog 0	Effective Inventory at wholesaler
(16)	Eff Inv 1 = Inventory 1 - Backlog 1	Effective Inventory at distributor
(17)	Eff Inv 2 = Inventory 2 - Backlog 2	Effective Inventory at factory
(18)	FINAL TIME = 36	The final time for the simulation.
(19)	in = DELAY FIXED(sold 0 , 2, 4)	Incoming orders at retailer
(20)	in 0 = DELAY FIXED(sold 1 , 2, 4)	Incoming orders at wholesaler
(21)	in 1 = DELAY FIXED(sold 2 , 2, 4)	Incoming orders at distributor
(22)	in 2 = DELAY FIXED(coming , 2, 4)	Incoming orders at factory
(23)	INITIAL TIME = 0	The initial time for the simulation.
(24)	Inventory = INTEG(in - sold , 12)	Physical inventory at retailer
(25)	Inventory 0 = INTEG(in 0 - sold 0 , 12)	Physical inventory at wholesaler
(26)	Inventory 1 = INTEG(in 1 - sold 1 , 12)	Physical inventory at distributor
(27)	Inventory 2 = INTEG(in 2 - sold 2 , 12)	Physical inventory at factory
(28)	ORDer = 4 + STEP (4, 5)	Weekly customer orders
(29)	ordered = DELAY FIXED(placed , 1, 4)	In transit orders by retailer
(30)	ordered 0 = DELAY FIXED(placed 0 , 1, 4)	In transit orders by wholesaler
(31)	ordered 1 = DELAY FIXED(placed 1 , 1, 4)	In transit orders by distributor
(32)	ordered 2 = DELAY FIXED(placed 2 , 1, 4)	In transit orders by factory
(33)	placed = MAX (0, SMOOTH (ORDer , SMOOTHTIME) + A * (12 - (Inventory - Backlog) - B * SupplyL))	Orders placed by retailer
(34)	placed 0 = MAX (0, SMOOTH (ordered , SMOOTHTIME) + A * (12 - (Inventory 0 - Backlog 0) - B * SupplyL 0))	Orders placed by wholesaler
(35)	placed 1 = MAX (0, SMOOTH (ordered 0 , SMOOTHTIME) + A * (12 - (Inventory 1 - Backlog 1) - B * SupplyL 1))	Orders placed by distributor
(36)	placed 2 = MAX (0, SMOOTH (ordered 1 , SMOOTHTIME) + A * (12 - (Inventory 2 - Backlog 2) - B * SupplyL 2))	Orders placed by factory
(37)	SAVEPER = TIME STEP	Frequency at which output is stored.
(38)	sFlow = placed - in	Supply line accumulation - retailer
(39)	sFlow 0 = placed 0 - in 0	Supply line accumulation - wholesaler
(40)	sFlow 1 = placed 1 - in 1	Supply line accumulation - distributor
(41)	sFlow 2 = placed 2 - in 2	Supply line accumulation - factory
(42)	SMOOTHTIME = 1	Forecasting parameter
(43)	sold = MIN (Inventory + in , ORDer + Backlog)	Crates sold by retailer
(44)	sold 0 = MIN (Inventory 0 + in 0 , ordered + Backlog 0)	Crates sold by wholesaler
(45)	sold 1 = MIN (Inventory 1 + in 1 , ordered 0 + Backlog 1)	Crates sold by distributor
(46)	sold 2 = MIN (Inventory 2 + in 2 , ordered 1 + Backlog 2)	Crates sold by factory
(47)	SupplyL = INTEG(sFlow , 0)	Supply line for retailer
(48)	SupplyL 0 = INTEG(sFlow 0 , 0)	Supply line for wholesaler
(49)	SupplyL 1 = INTEG(sFlow 1 , 0)	Supply line for distributor
(50)	SupplyL 2 = INTEG(sFlow 2 , 0)	Supply line for factory
(51)	TIME STEP = 1	The time step for the simulation.

Table 3.1 Equations simulating the traditional beer game

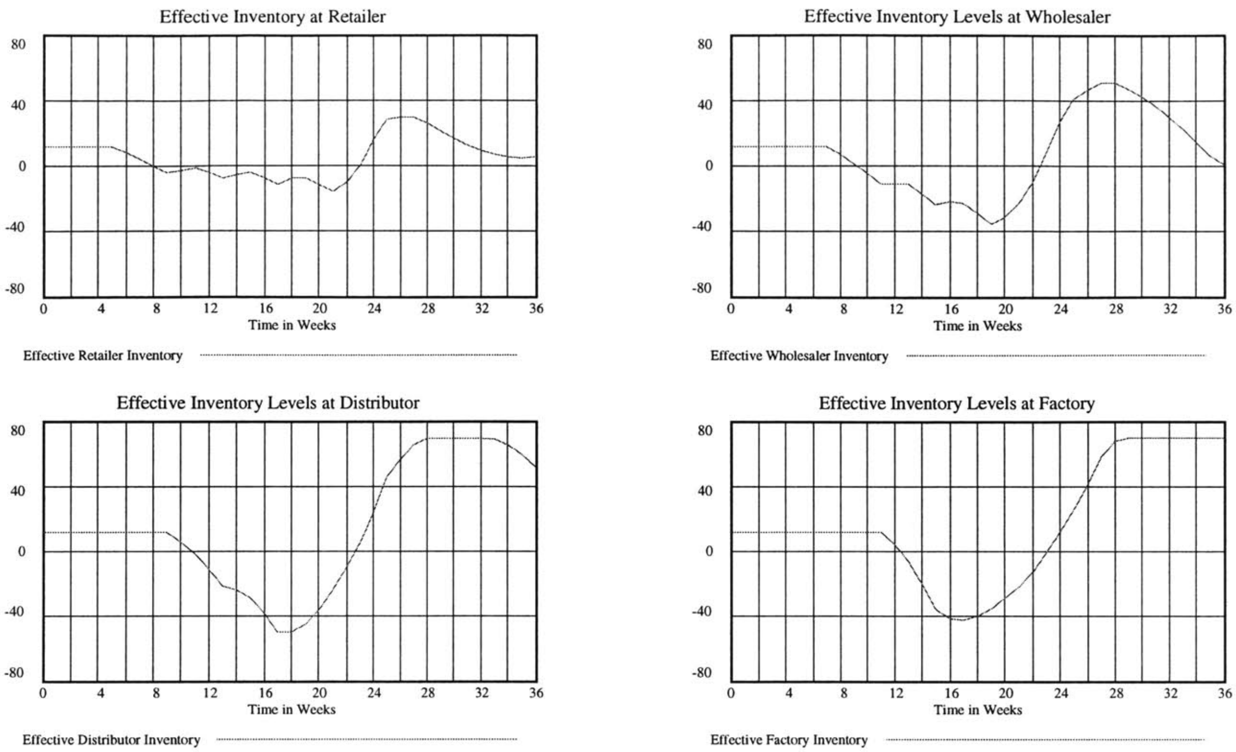


Figure 3.4 Effective inventory levels in the traditional supply chain

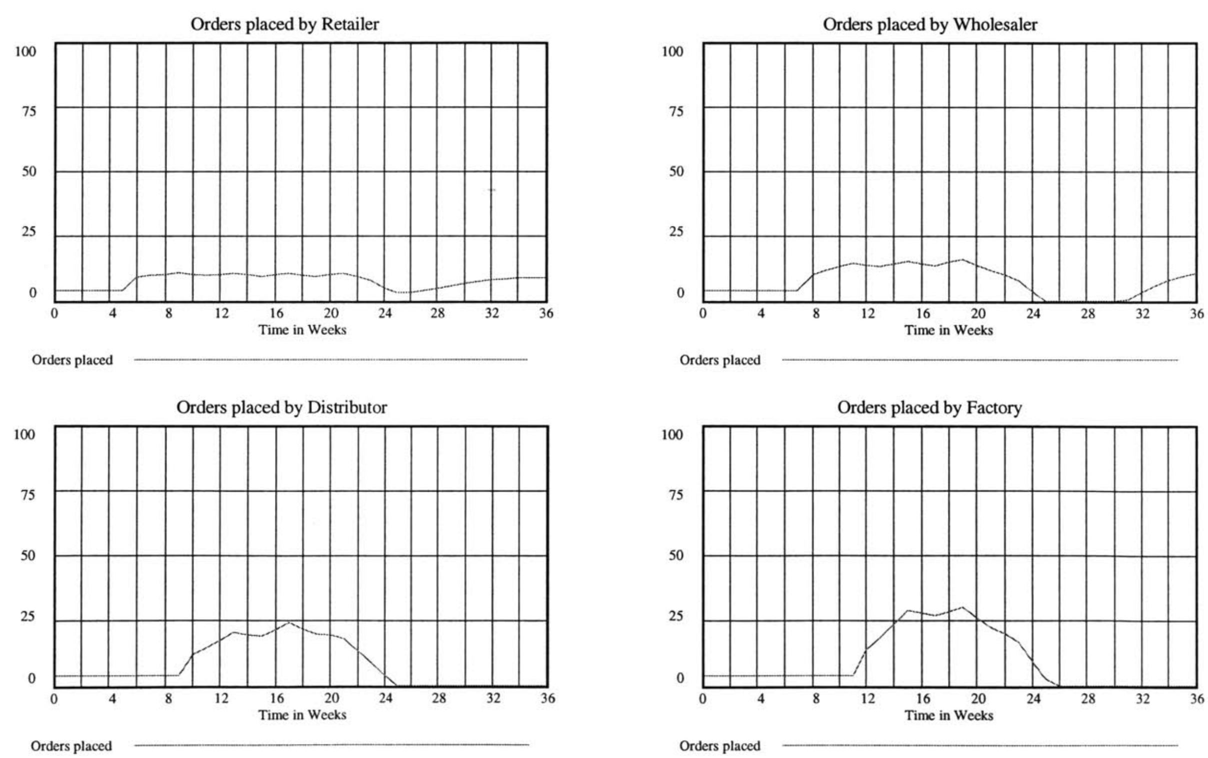


Figure 3.5 Orders placed by units in the traditional supply chain.

From Figure 3.4, we observe that effective inventory at the retailer, which is the sum of physical inventory and backlog, becomes zero first, and then negative. This pattern is repeated at the wholesaler, distributor, and the factory with a time delay and amplification of magnitude. This also has an affect on the orders placed by units every week, as observed in Figure 3.5. We see a surge in the orders placed as the weeks pass by, and this surge increases from retailer to wholesaler to distributor to factory. This pattern of demand amplification up along the supply line is regularly observed in the beer game trials, and also in many industrial settings. Procter and Gamble (P&G) first observed this pattern in their Pampers business. While the consumer demand for Pampers was fairly constant, they observed that the demand as seen by wholesalers had a large variation. And by the time the demand reached P&G, the fluctuations were even higher. They named this phenomenon the "bullwhip effect" [34]. Hewlett Packard has also observed demand amplification and distortion in their printer supply chain [34a]. The bullwhip effect distorts demand information as it passes up along the supply chain. It causes cycles of excessive inventory and severe backlogs, poor product forecasts, unbalanced capacities, poor customer service, uncertain production plans, and high backlog costs, or sometimes even lost sales.

Lee, *et al.* [34] have given more examples of industrial supply chains suffering from the bullwhip effect, and analyzed its probable causes. Delay in passing information from one unit to another is a primary cause. Another important cause is independent demand forecasting within each unit with emphasis on its customer – retailers forecast demand for customers, wholesalers forecast demand for retailers, etc. A third cause is the unavailability of downstream / upstream information to other units in the supply chain, causing them to take decisions on limited available data. These constitute lack of information visibility, along with lack of collaboration efforts amongst units in the supply chain.

In the next section, we simulate the beer game under different conditions to study the effect of information visibility and lack of collaboration on its dynamics. We simulate it under two different forecasting policies. These policies are based on forecasting functions built-in in Vensim. We then simulate the beer game without and with information visibility and collaboration. We present the supply chain dynamics observed under each of these situations, and compare supply chain costs.

3.4 Beer game model simulations

Following are the different simulation scenarios for the beer game, intended to produce varying system dynamics:

I. Using FORECAST function for demand forecasting, without information visibility or collaboration.
II. Using SMOOTH function for demand forecasting, without information visibility or collaboration.
III. Using FORECAST function for demand forecasting, with information visibility and collaboration
IV. Using SMOOTH function for demand forecasting, with information visibility and collaboration.

The FORECAST function in Vensim provides a trend extrapolation forecast of the future value of a variable based on its past behavior. The SMOOTH function provides a exponential smoothing of input depending on delay time input to the function. Values for forecasting parameters suggested in [28] are based on extensive observation of the data from many beer games played in the past. We ran sample runs of the simulation to converge on optimal values for the forecasting function parameters.

In the traditional beer game, end customer demands are only visible to the retailer. Also, there is a 2-week delay in passing order requests from one unit to another. Furthermore, inventory levels and backlogs at individual units are cut-off from each other. Each unit forecasts its demand. These constitute a lack of information visibility and collaboration. Providing information visibility and collaborating amongst units implies elimination of the above mentioned delays in data communication, availability of end customer demand to all units, and a sharing of unit-specific information such as inventory levels, planned purchases, supply line levels, etc. In case III and IV, we build new models of the beer game from first principles to incorporate these improvements in collaboration and information sharing. In these simulations, the customer demand is made visible to all the four units in the supply chain directly in real time, and is not restricted only to the retailer. Although not used in the policy for taking forecasting or shipping decisions, also available to all units is information about inventory and backlog levels at each of the units.

We present below the analysis for each of the four cases described above. For each case, we first present the Vensim model used to simulate the beer game. As in the previous section, the model diagram depicts all the variables used in the model, and also the interrelations between variables. The detailed equations describing these variables are listed in tables in Appendix A. The supply chain dynamics associated with each model are presented in graphs for effective inventory at each unit during the time period of simulation, and the orders placed by each unit. We also calculate the total inventory costs associated with each case, and compare them in Figure 3.18.

The forecasting technique used for ordering beer crates from other units in the supply chain with information visibility is similar to the one described earlier; except that it makes use of global information visibility. It accounts for current customer demand, and forecasts demand a few weeks ahead as required. Direct use of customer demand alone has yielded us good results. As a future exercise, we can make available information about inventory levels and backlogs at each unit for decisions being made about how many crates should be shipped or how much beer should the factory manufacture next; and study supply chain dynamics under changed visibility and ordering policy conditions.

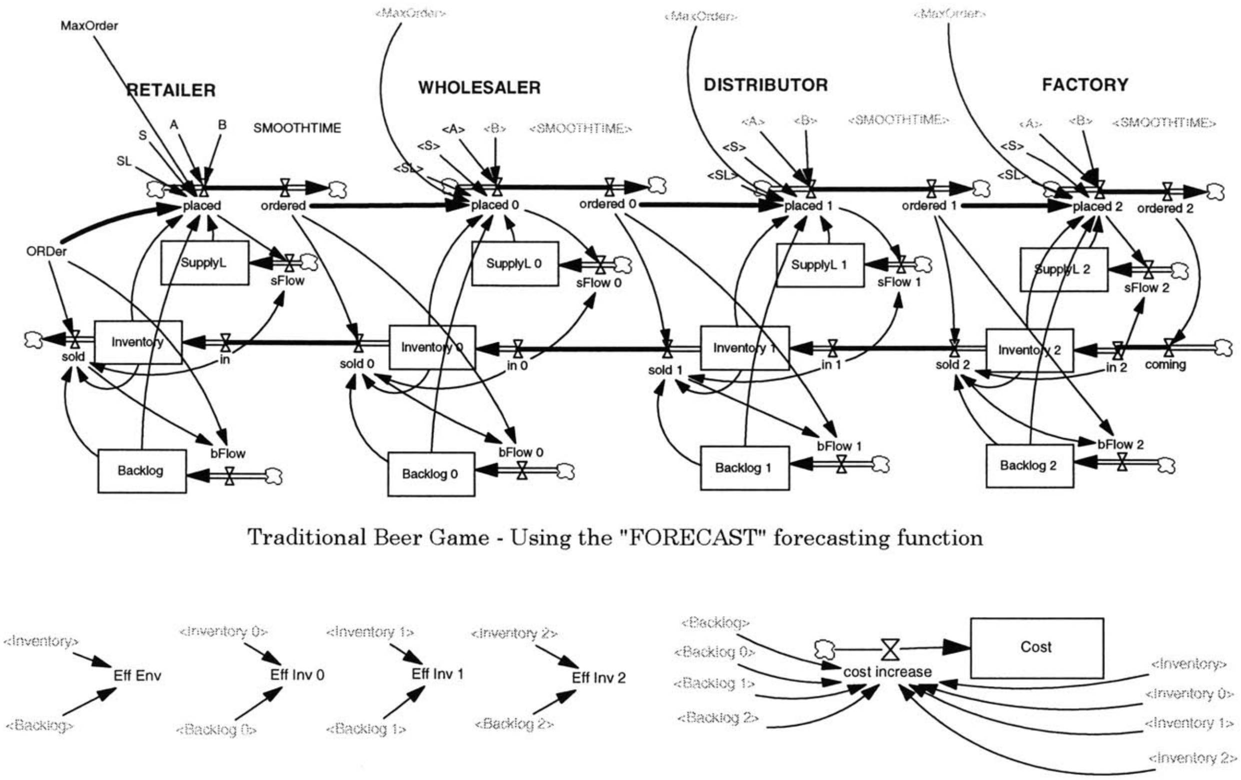


Figure 3.6 Beer game simulation Case I.

Case I

Information Visibility and Collaboration amongst units: *NO*

Forecasting function: *FORECAST*

Behavior:
The bullwhip effect is very strong in this case. Changes in orders placed are sharp and rapid. Orders vary from (4 to 35) for the retailer. This amplifies to (0 to 100) for the factory. Inventory levels show massive swings from (0 to 25) for the retailer to (–75 to 175) for the factory. System behavior does not reach a steady state, the factory inventory is hovering high at about 150 at the end of the simulation.

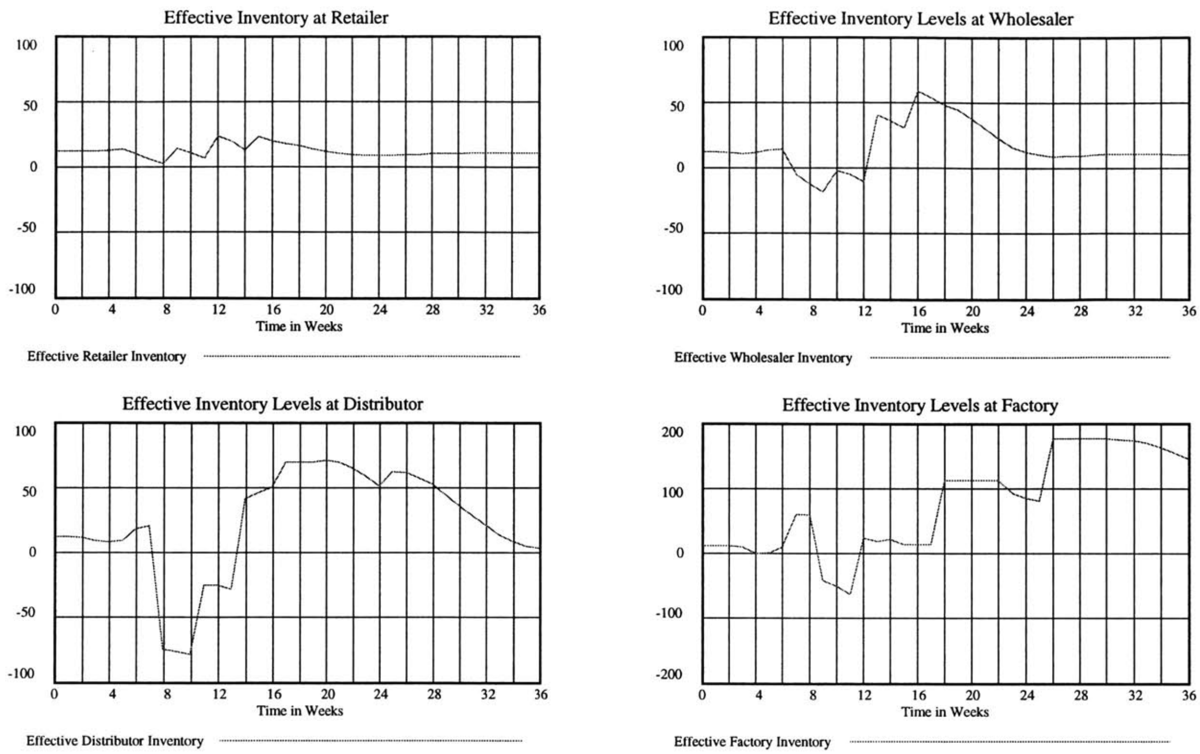


Figure 3.7 Effective inventory levels for Case I

(Note: The scale on the graph for Effective Inventory Levels at the Factory is –200 to 200!)

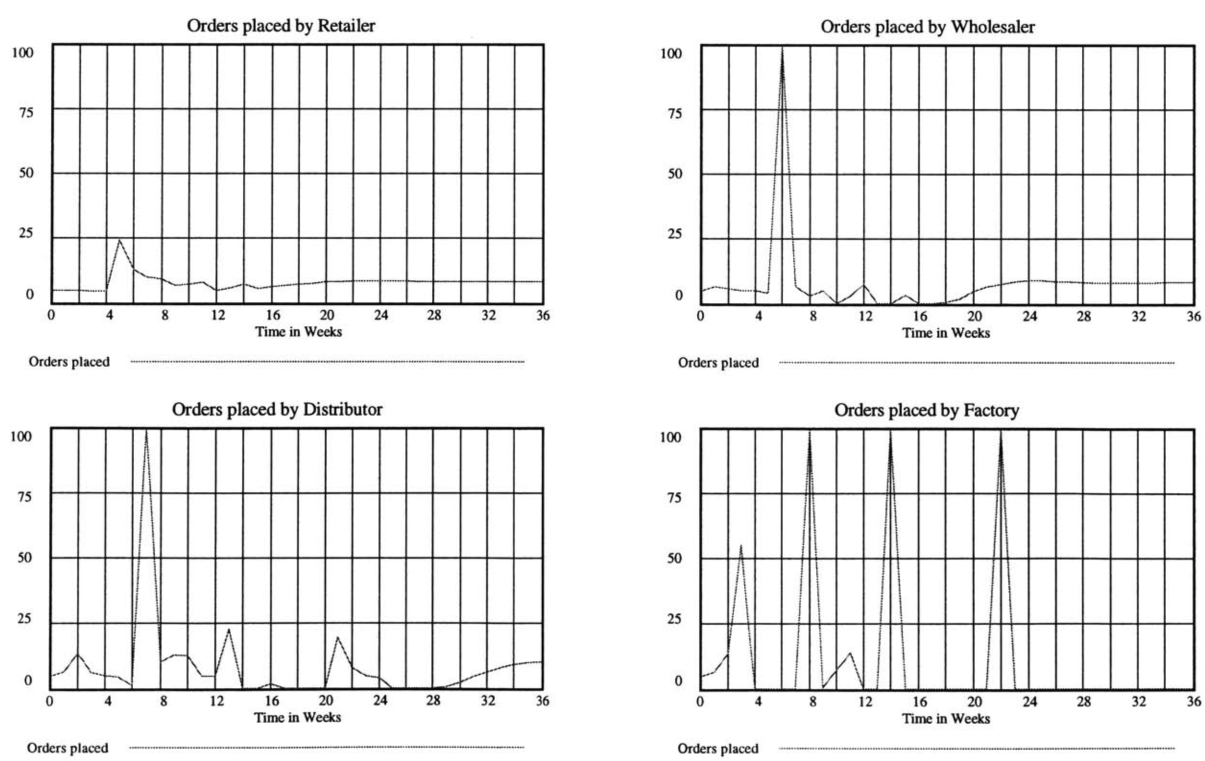
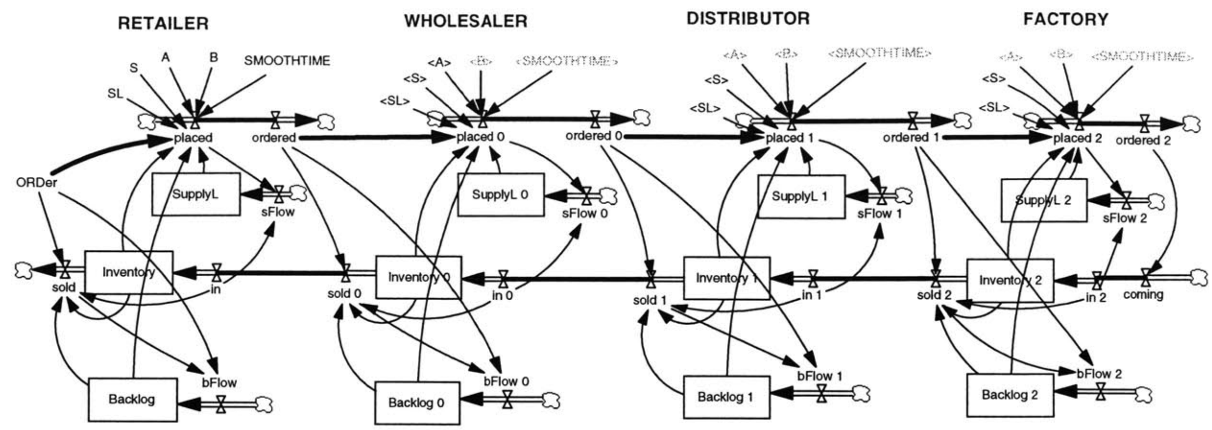


Figure 3.8 Orders placed for Case I



Traditional Beer Game - Using the "SMOOTH" forecasting function.

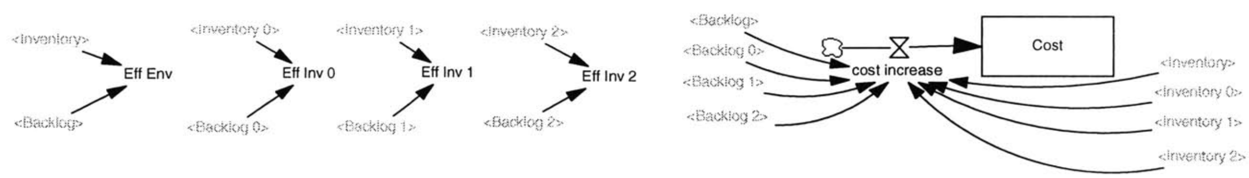


Figure 3.9 Beer game simulation Case II.

Case II

Information Visibility and Collaboration amongst units: *NO*

Forecasting function: *SMOOTH*

Behavior:
The bullwhip effect is strong in this case too. The forecasting function behaves bettering this case, and shows smooth behavior compared to the one in the previous case. Order variation at the retailer is from (4 to12) and it amplifies to (0 to 30) at the factory. Inventory levels swing from (0 to 15) at the retailer and (–35 to 60) at the factory. At the end of 36 weeks, the inventory levels at distributor and factory haven't yet achieved steady state.

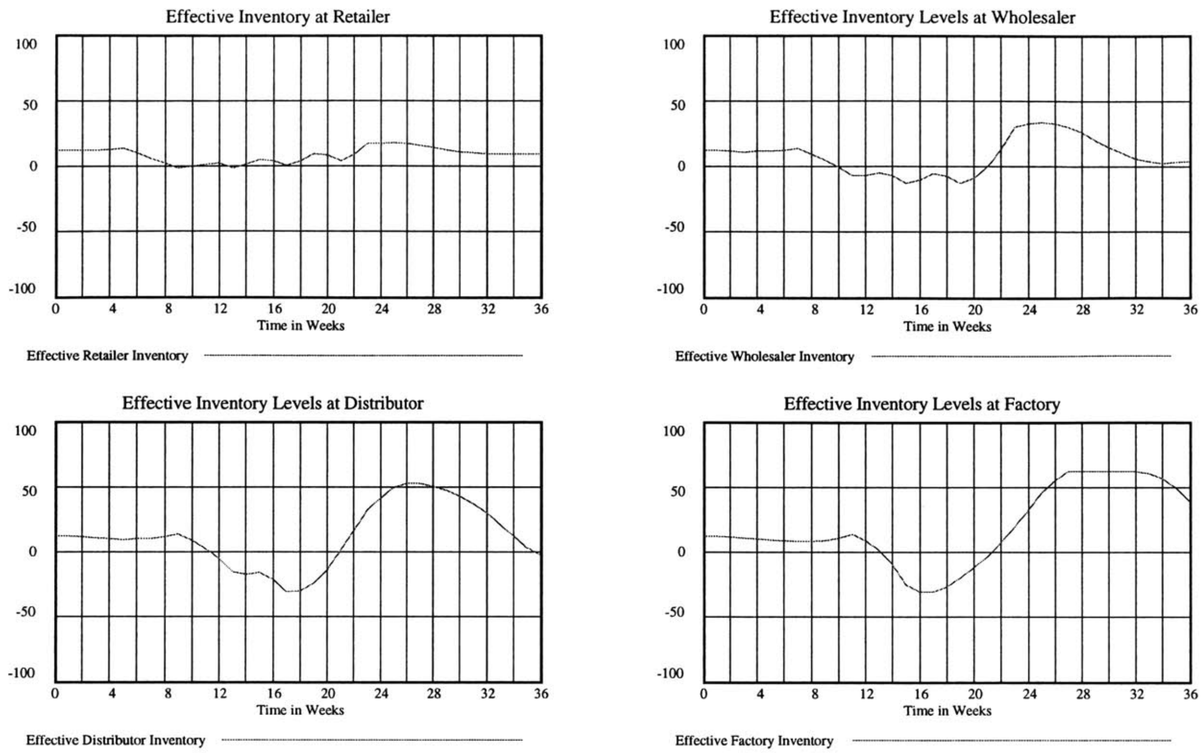


Figure 3.10 Effective inventory levels for Case II

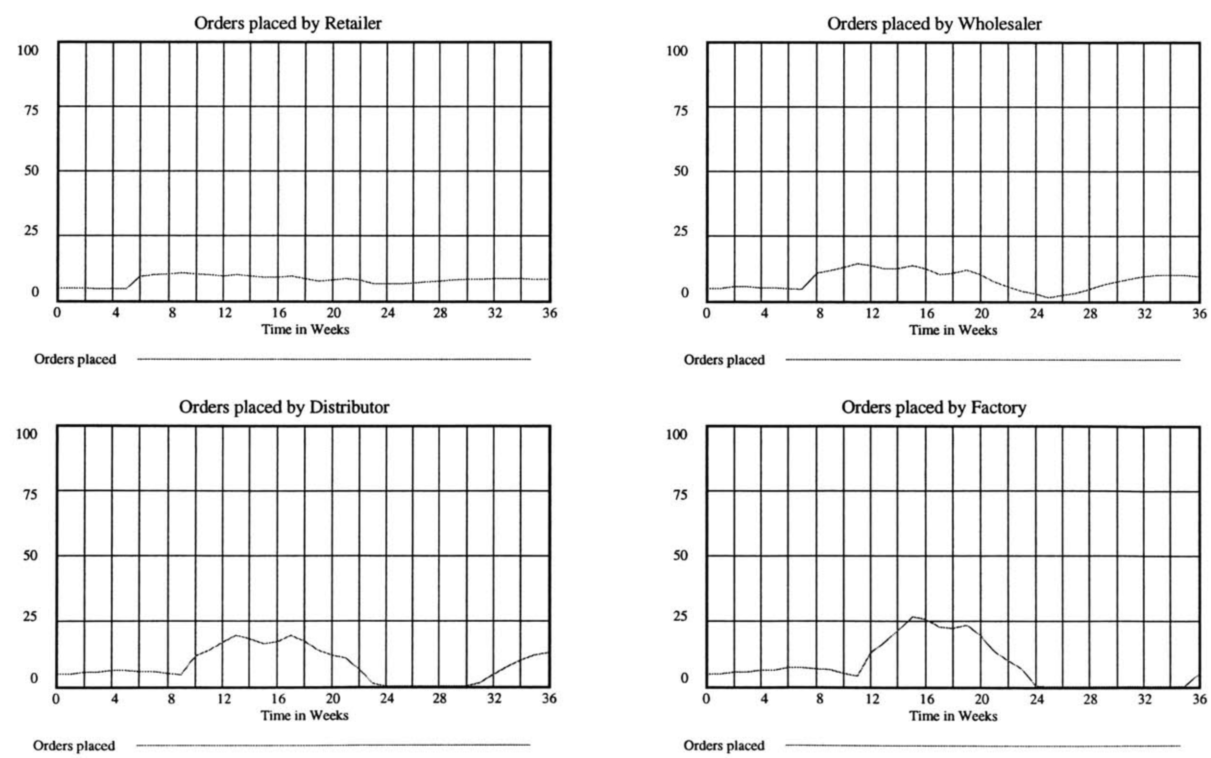


Figure 3.11 Orders placed for Case II

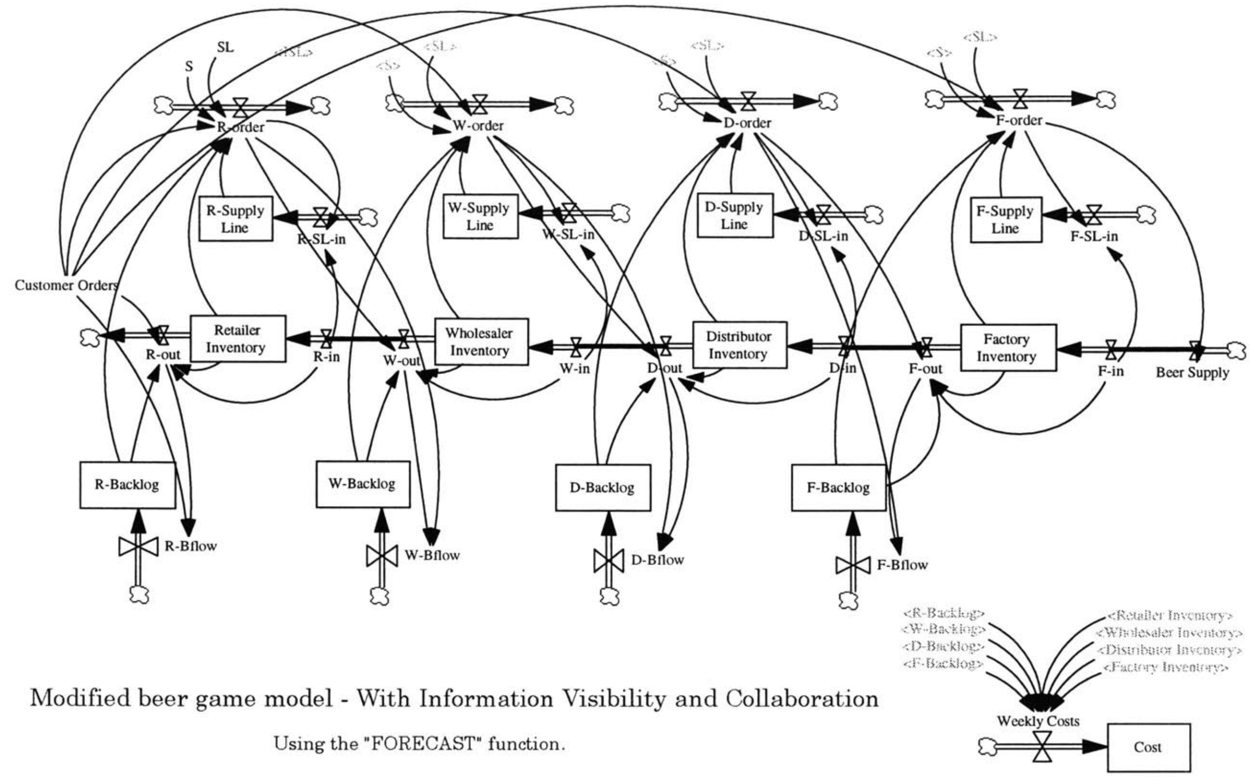


Figure 3.12 Beer game simulation Case III

Case III

Information Visibility and Collaboration amongst units: *YES*

Forecasting function: *FORECAST*

Behavior:
This case shows less variation than the previous two. Here, customer demand is made visible to all units in the supply chain. Decisions to ship out crates are also taken centrally. Here, order variation at the retailer is from (0 to 25) and pretty much the same at all the other units in the supply chain. Effective inventory varies from (12 to 25) at the retailer and (0 to 40) for the factory. All the units achieve steady state in this case.

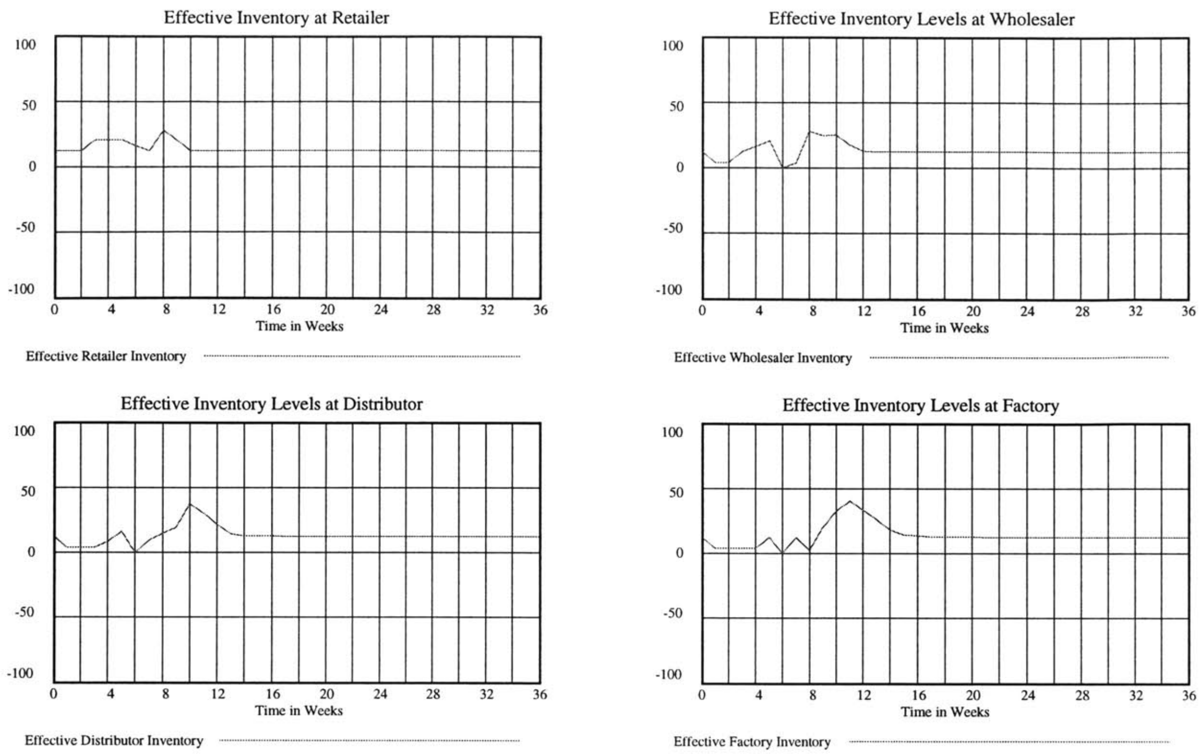


Figure 3.13 Effective inventory levels for Case III

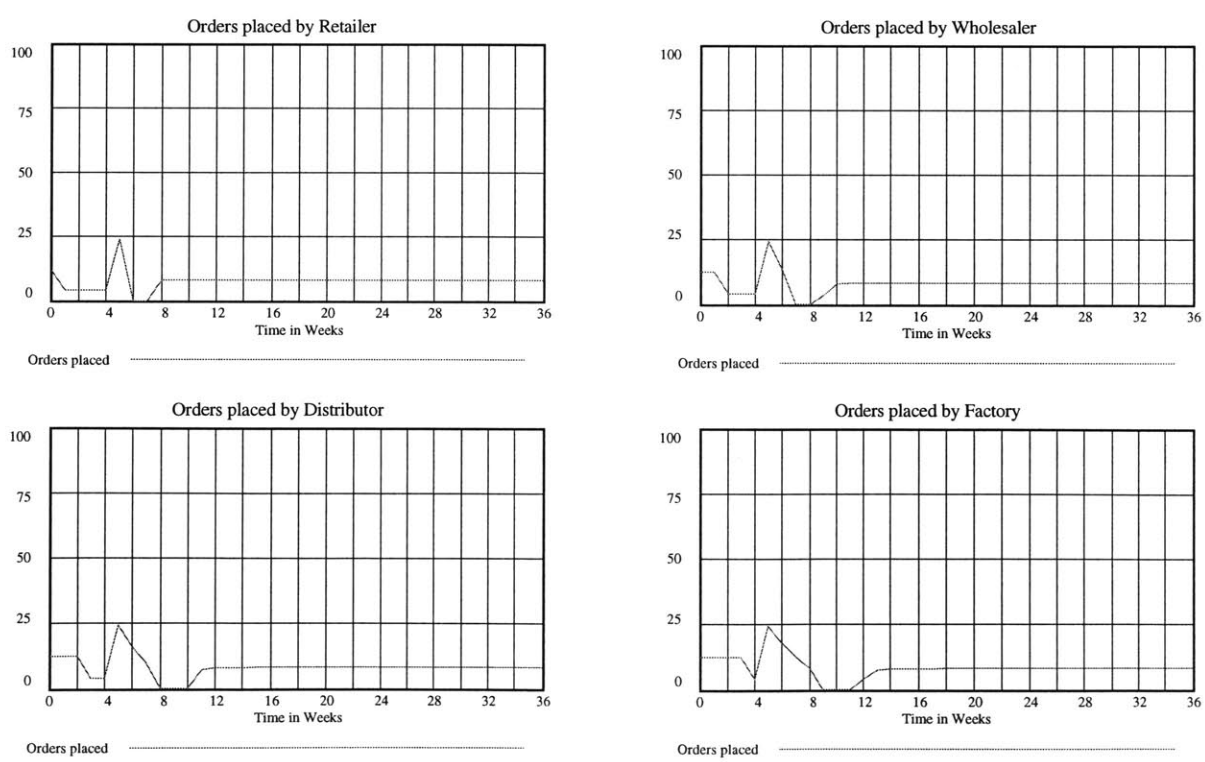


Figure 3.14 Orders placed for Case III

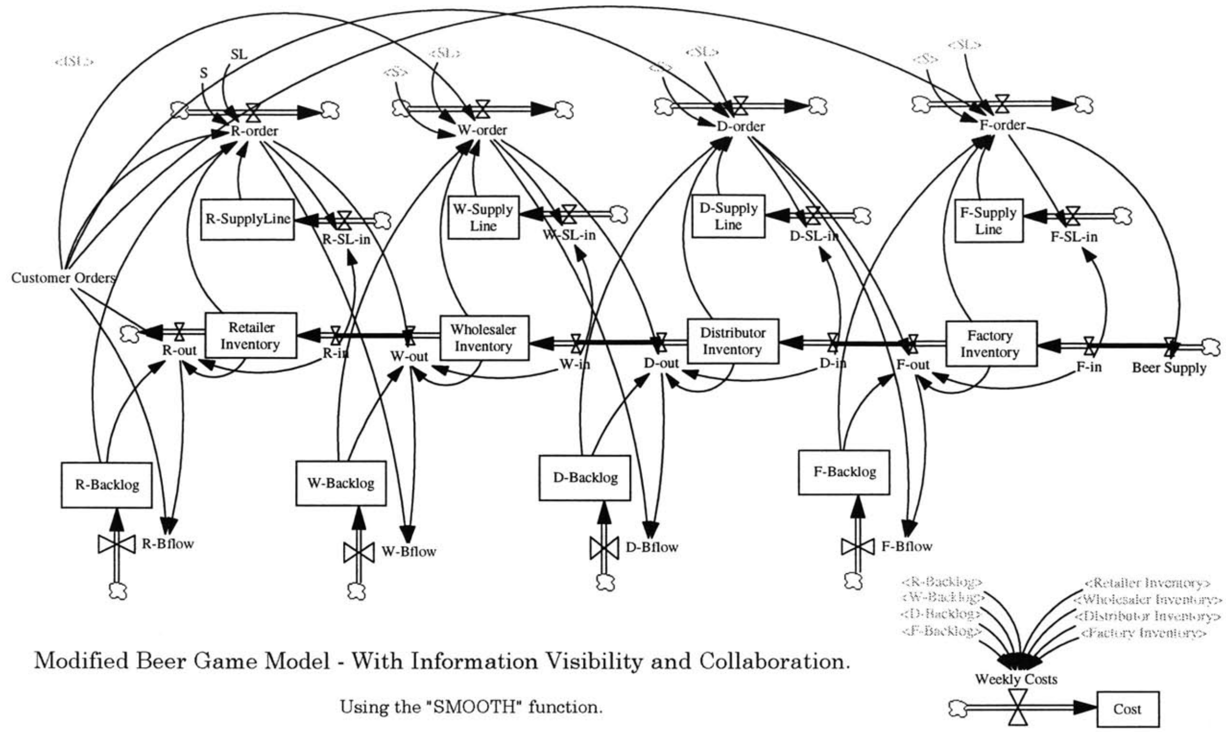


Figure 3.15 Beer game simulation Case IV

Case IV

Information Visibility and Collaboration amongst units: *YES*

Forecasting function: *SMOOTH*

Behavior:
This case has the least variations. The variation in orders is (4 to 12) for all units in the supply chain. Effective inventory varies from (8 to 20) for the retailer and (4 to 25) for the factory. All units achieve steady state behavior in this case.

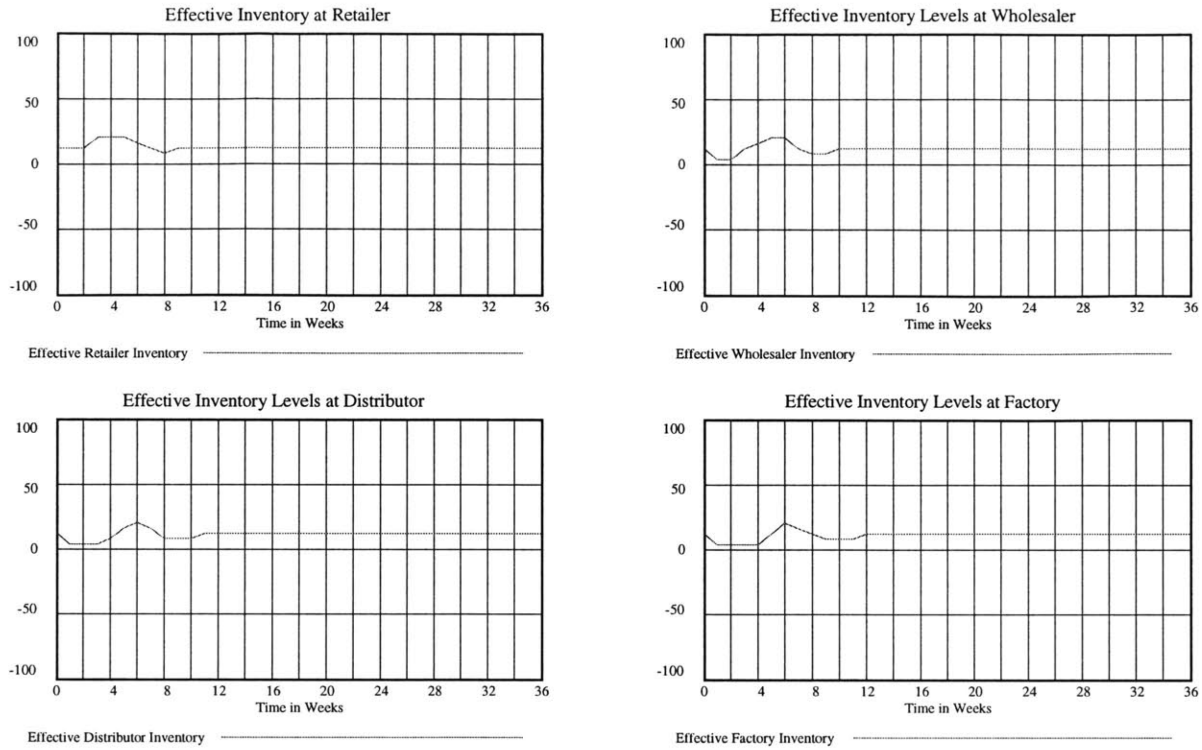


Figure 3.16 Effective inventory levels for Case IV

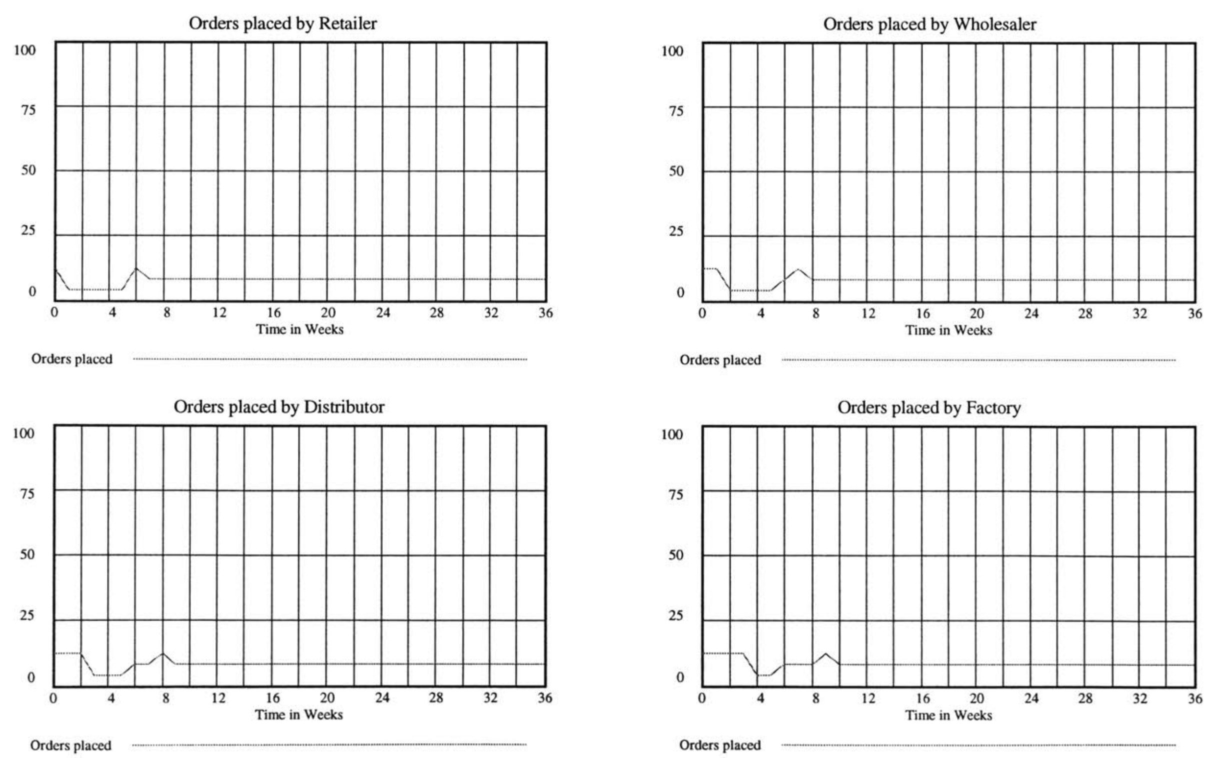


Figure 3.17 Orders placed for Case IV

3.5 Explanation of simulation results

Case	Information Visibility and Collaboration	Forecasting Function	Cost $
I	No	FORECAST	3020
II	No	SMOOTH	1482
III	Yes	FORECAST	957
IV	Yes	SMOOTH	852

Table 3.2 Comparison of the four cases.

Figure 3.8 and 3.11 clearly show the bullwhip effect in the supply chain in the absence of information visibility. Figure 3.10 and 3.11 also show the amplification of fluctuations in demand and inventory levels from retailer to factory. We observe that the

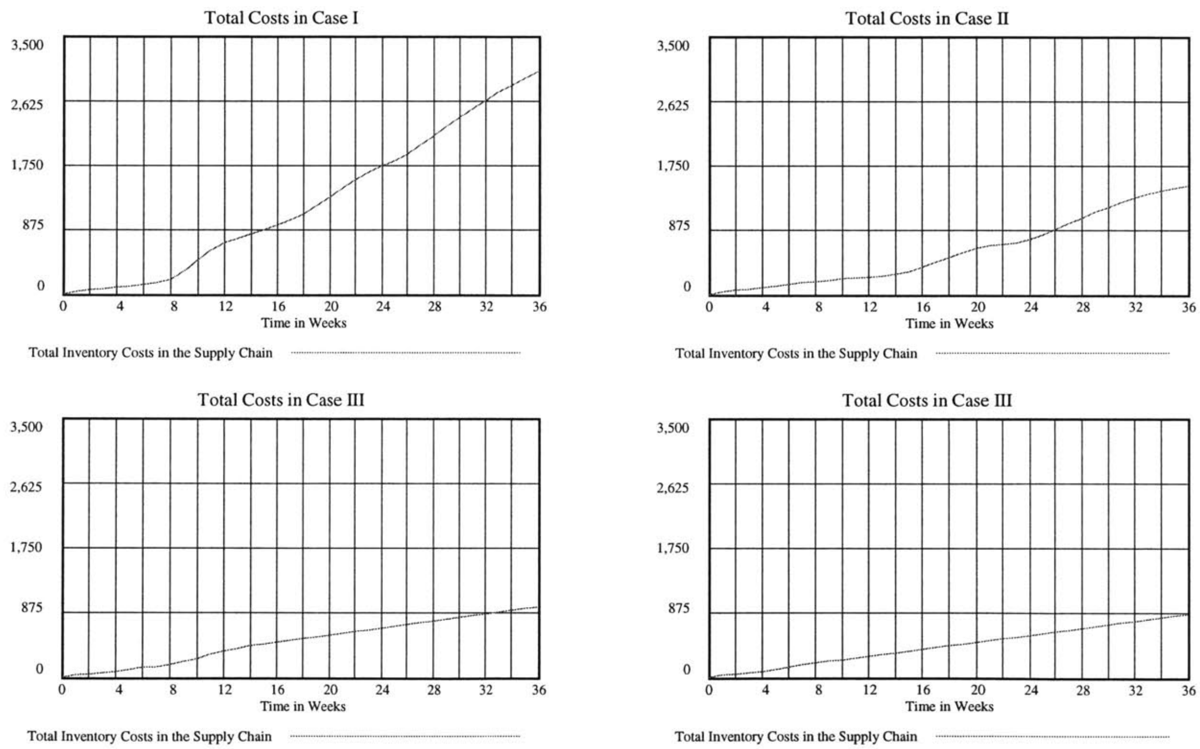


Figure 3.18 Comparison of costs in the four cases.

bullwhip effect is more pronounced in case I than case II. This is an outcome of the forecasting function used in the two cases. The "FORECAST" function used in case I changes rapidly with changes in inputs. These get amplified very easily as demand forecasting progresses upstream in the supply chain. On the other hand, the "SMOOTH" function used in case II changes smoothly despite rapid changes in inputs – creating a smooth continuous effect, resulting in well-behaved ordering patterns.

The behavior of the supply chain in cases III and IV (where we achieve information visibility of customer demand at each unit) is a great improvement over those in cases I and II. We see that information visibility and collaboration has almost eliminated the bullwhip effect. In case III and IV, we provided visibility by making available customer demand data to all units, and eliminating communication delays between units by assuming instantaneous communication. This elimination of delays and collaborated forecasting of demand has clearly eliminated the ripple effect in inventory levels at or orders placed by individual units. We observe that in cases of improved information visibility, the effective inventory has not fallen into backlog, and has consistently delivered the goods demanded by the customer. We also observe that the "SMOOTH" function is a better forecasting technique than the "FORECAST" function for this example of the beer game. Given the extremely complex nature of supply chains, with many inherent interdependencies, and external influencing factors, different forecasting techniques might yield different results. Although given these complexities, an important conclusion falls out of the above simulations while comparing results of case I with case III; and case II with case IV. We see that for a given forecasting technique, information visibility and collaboration offers a huge savings (~ 40% to 70%) in inventory costs in the supply chain.

Information visibility in a supply chain aims at having the right data at the right place and the right time in the right format – for ex: customer demand for crates of beer per week, inventory and backlog levels in each unit, etc. An initiative to provide information visibility can be implemented as primarily a combination of two important sub-initiatives: having the technology required for achieving real time data management; and willingness of supply chain units for sharing their data with trading partners and collaborate in the supply chain. We discussed the latter initiative – of planning, forecasting and replenishing demand collaboratively – in the previous chapter. We also talked about current technological means for achieving information visibility and their limitations. In the next chapter, we describe a framework for achieving information visibility in supply chains that can overcome most of these limitations.

Chapter 4

A framework for information visibility

4.1 Introduction

In the previous chapters, we saw the pronounced need for visibility of information in the supply chain. All software reviewed in Chapter 2 operates successfully if correct information is available to it. In Chapter 3, we saw the improvements in supply chain costs offered by information visibility. In this chapter, we propose a framework by which we can achieve information visibility in real time.

4.2 Information visibility in real time

As explained in Chapter 2, most data that is gathered today is not real time. Data is gathered using various techniques – manual, barcodes or tags. Different people gather it at different times of the day in individual units of the supply chain, on the shop floor or in the warehouse. This data is then entered in the computerized database at a predetermined time, at regular intervals. Thus, the database is updated in fixed quanta of time. In between these time periods, which typically can last from a few hours to a few days, the database is only pseudo-real time. This introduces visibility gaps in the system. In many instances, other units in the supply chain base their decisions on data that they gather from this unit under consideration. They are provided with inaccurate data, which results in sub-optimal decision-making.

To achieve real time visibility of information, we need to:

1. Have real time data acquisition mechanisms.
2. Convert acquired data into relevant information: using standardized, secure representation.
3. Have instantaneous access to this information.

We propose the use of automated tracking systems to achieve real time data acquisition. The purpose of an automated tracking system is to capture relevant data about the object under consideration and pass it on to another application. The requirements of such a system include:

- A mechanism to locate the object.
- A mechanism to gather relevant data.
- Interface with other applications to exchange this data.

We propose the use of radio frequency tags, in conjunction with tag readers to achieve real time data acquisition.

Data acquired by tag readers will then be passed on to a software application. This application, in conjunction with a lookup table and object description language, process the data and convert it into relevant information, and store it at the appropriate location so that others can utilize it to take sensible decisions.

Instantaneous information access can be achieved through the internet.

4.3 Basic building blocks

The basic building blocks of the framework outlined above are radio frequency tags, electronic product codes, tag readers, object naming service, an object description language, the internet, and a controlling software application.

4.3.1 Electronic Product Code (ePC)

The encoded bits on the radio frequency tag will constitute an object identification code, called "electronic Product Code", or in short, the ePC. Brock [35] invented the ePC at the MIT Auto-ID Center. Brock defines ePC as "a numbering scheme that can provide unique identification for physical objects, assemblies and systems." The code acts as a pointer to another location where information is stored remotely about the object being identified by the code.

ePC is being developed as a replacement for the fast depleting UCC (Uniform Code Council) barcode, and is aimed to be the next generation object identifier. The ePC primarily consists of bits providing a unique identification to every object. The ePC was initially defined in Foley, *et al.* [36a]. More details about the ePC can be found at [36].

4.3.2 Tag readers

Given that every object is labeled with a unique ePC, the next task is to identify them. This is done with the help of radio frequency tag readers. Foley, *et al.* [36a] define tag readers as "a microcontrolled unit with an output coil wound for a particular carrier frequency. It consists of peak detector hardware, comparators, and firmware for doing backscatter communications." The tag reader generates a radio frequency sine wave that transmits energy to the tag, and receives data from it.

All points of data acquisition will be equipped with tag readers, capable of reading the ePC on tags within their read range. Tag readers can currently read within a range of 1-3 meters. Further developments in tag reading technology will hopefully extend the space covered by a single tag reader. Also, anti-collision technology enables reading of more than one tag within a given readers range (up to as much as 50 per second currently [38], and more with time). As objects come into the reading zone, the tag reader reads the object ePCs. Once the tag reader has received bits from the object, the bits are arranged into a sequence of numbers, which constitutes the unique ePC for the object being identified.

4.3.3 Object Naming Service (ONS)

ePC received from the tag reader is passed on to a protocol that will help determine where to locate information corresponding to the ePC under consideration. This protocol is called "Object Naming Service" (ONS), and is being developed at the Auto-ID Center by Sarma, *et al.* [37]. The task of the ONS is to point to the source of query the destination where information relevant to the ePC being queried is stored. Information about objects will be stored at convenient locations on the internet. For example: information about an object, say soap X, can be stored on the website of company Y, which manufactures soap X. Anyone wishing to know more about soap X can access that information from the ePC of soap X. The company database could contain such information about soap X as the constituents of the soap, skin type serviced, manufacturing date and location, shipping details, etc. Different parts of this information would be relevant to different persons coming in contact with soap X, for different purposes, all along the supply chain for soap X, from the manufacturer to the consumer.
The ONS would serve the purpose of connecting the source of query to this information on the internet.

4.3.4 Object description language (ODL)

The location to which the ONS points for information about a particular object is an IP (internet protocol) address. At this IP address, information about the object is represented. Since no units in the supply chain communicate solely within itself, but also with other units in its supply chain, the language used for this representation should satisfy requirements like:

- Possess a standardized, open format.
- Be inter-operable.
- Be scalable.
- Provide required security features.
- Be robust and manageable.
- Be easy to implement.

The extensible Markup Language (XML) lends itself easily to satisfy many of the above criteria. XML is a "meta-language" that defines syntax rules for "well-formed" documents, and define an unlimited number of languages for specific industries and applications. It has rapidly become the standard for defining data interchange formats.

Many independent committees are working towards defining standard practices for the use of XML. The Techniques and Methodologies Working Group (TMWG), a permanent working group under the United Nations Center for Facilitation of Procedures and Practices for Administration, Commerce and Transport (CEFACT) has come out with specific recommendations for converting current standard EDI syntax into XML syntax [39]. The World Wide Web Consortium (W3C) is involved in developing core XML technologies and other standards related to XML [40]. The Organization for the Advancement of Structured Information Standards (OASIS), a vendor neutral international consortium, focuses on making XML standards practical and easy to adopt [41]. A recent OASIS spin-off is XML.ORG, an industry web portal, and aims at establishing an open, distributed system for enabling the use of XML in electronic commerce and other industrial applications. XML.ORG is designed to provide a credible source of accurate, timely information about the application of XML in industrial and commercial settings and to serve as a reference repository for XML specifications such as vocabularies, document type definitions (DTD), schemas, and namespaces [42].

Recently, UN/CEFACT and OASIS have joined forces to initiate a worldwide project to standardize XML business specifications. UN/CEFACT and OASIS have established the Electronic Business XML (ebXML) initiative to develop a technical framework that will enable XML to be utilized in a consistent manner for the exchange of all electronic business data [43]. The ebXML initiative intends to develop relevant and open technical specifications to support domestic and international electronic business exchanges.

The Auto-ID Center at MIT is developing the Product Markup Language (PML) as a "standard language" for describing physical objects [44]. Based on XML, PML will build layers of increasingly specific data describing physical objects. The objective of developing PML, as defined by the Auto-ID Center, are:

- should translate or contain static data - such as dosage, shipping, expiration, advertising and recycling information.
- should provide instructions for machines that "process" or alter a product - such as microwaves, laundry appliances, machine tools and industrial equipment.
- may need to communicate dynamic data: information that changes as a product ages or as it is consumed - such as volume, temperature, moisture and pressure.
- may need to include software, or programs, which describe how an object behaves - for instance: a .pml file may contain the program which describes how fast the tires on your car will wear before they need to be replaced, or how fast an object may burn in case of a fire.

4.3.5 Fitting them together

The ePCs on tags, coupled with tag readers and the controlling software application form an automated tracking system for real time data acquisition. Along with ONS, an appropriate object description language, and the internet, we can convert acquired data into relevant information and store it securely at appropriate locations. We explain this framework in detail in the next section.

4.4 A generalized framework for information visibility

We describe the framework with the help of an example – the distributor in the supply chain for beer.

4.4.1 Framework within an individual supply chain unit

The distributor receives beer crates from his supplier, which is the factory, and he ships out crates to his customer, which is the wholesaler. In real life, a distributor will typically receive more than one product from more than one factory, and will ship a mix of products to multiple wholesalers, or directly to retailers. The framework proposed below is highly scalable and can be easily extended to include the case of multiple products, multiple suppliers and multiple customers.

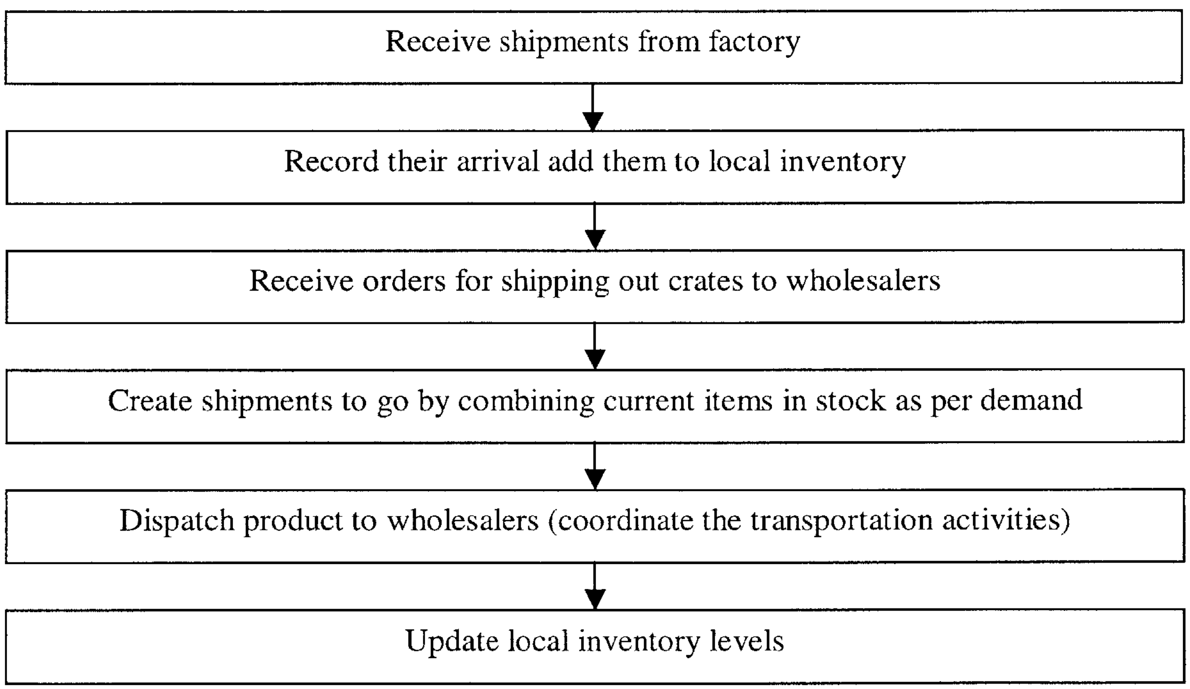


Figure 4.1 Activities flowchart at the distributor.

The distributor typically operates out of a large warehouse. Crucial tasks associated with running this distribution center are shown in Figure 4.1.

Implementing a tracking system for real time data acquisition would require that:

- All beer crates have radio frequency tags attached to them
- All tags carry an ePC on them
- The shipment receiving station be equipped with tag readers

- The inventory shelves be equipped with tag readers
- The dispatch location be equipped with tag readers

Data acquisition begins from the point when goods arrive at the warehouse. At the receiving dock, crates are offloaded from the trucks and put on to the conveyor belts receiving them into the warehouse inventory. Tag readers at the entry point, which read the ePCs on the tags, surround these conveyor belts. The crates then proceed to inventory shelves, which are also equipped with tag readers. When orders from customers are processed, and packages are ready to be shipped, they are passed out on to the loading dock through a set of tag readers again which note the ePCs of the crates being shipped. This is shown schematically in Figure 4.2. A software application managing data acquisition for the distributor warehouse communicates with the tag readers installed at various locations in the warehouse. It receives large amounts of data from many inputs. This data has the potential to be converted into real time information. The software application gathers the ePCs of products being tracked by the tag readers, queries the ONS for the web address where information about the product is stored, updates that information if necessary, plus maintains localized data records relevant to operations of the warehouse. The updated product information can be stored either in a centralized database or distributed databases.

For example: when a beer crate comes in through the receiving docks of the warehouse, it passes through the tag readers at the incoming station. The tag reader reads the ePC of the tag on this beer crate and passes this information to the software application managing the data acquisition process. The software application, in conjunction with the ONS, locates the address at which detailed information about this crate is stored. Since the software application also knows the location of the tag reader from which it has received information about the beer crate, it is in a position to update location information relevant to this beer crate. With requisite security features and access control, the data acquisition software can now access information about this beer crate, as well as update the current status of the crate, and maintain real time data accuracy. The beer crate is then stored in inventory at the warehouse – typically on shelves. We equip each of these shelves with a tag reader. The software application continuously receives information from these tag readers about the ePCs of the crates stored on the inventory shelves. If the application records a new ePC, it immediately coordinates with the ONS to reach the information site of this crate, and learn information about the crate from that site. It can again update the information about this crate – about its current location, time, and destination and waiting period if known. Provided the waiting period for a crate is known, the software application can alert the warehouse managers at a suitable time before the end of the period about further shipping actions for that crate. The software application can also perform other routine tasks, such as issuing warnings if the crate is reaching its expiry date, etc.

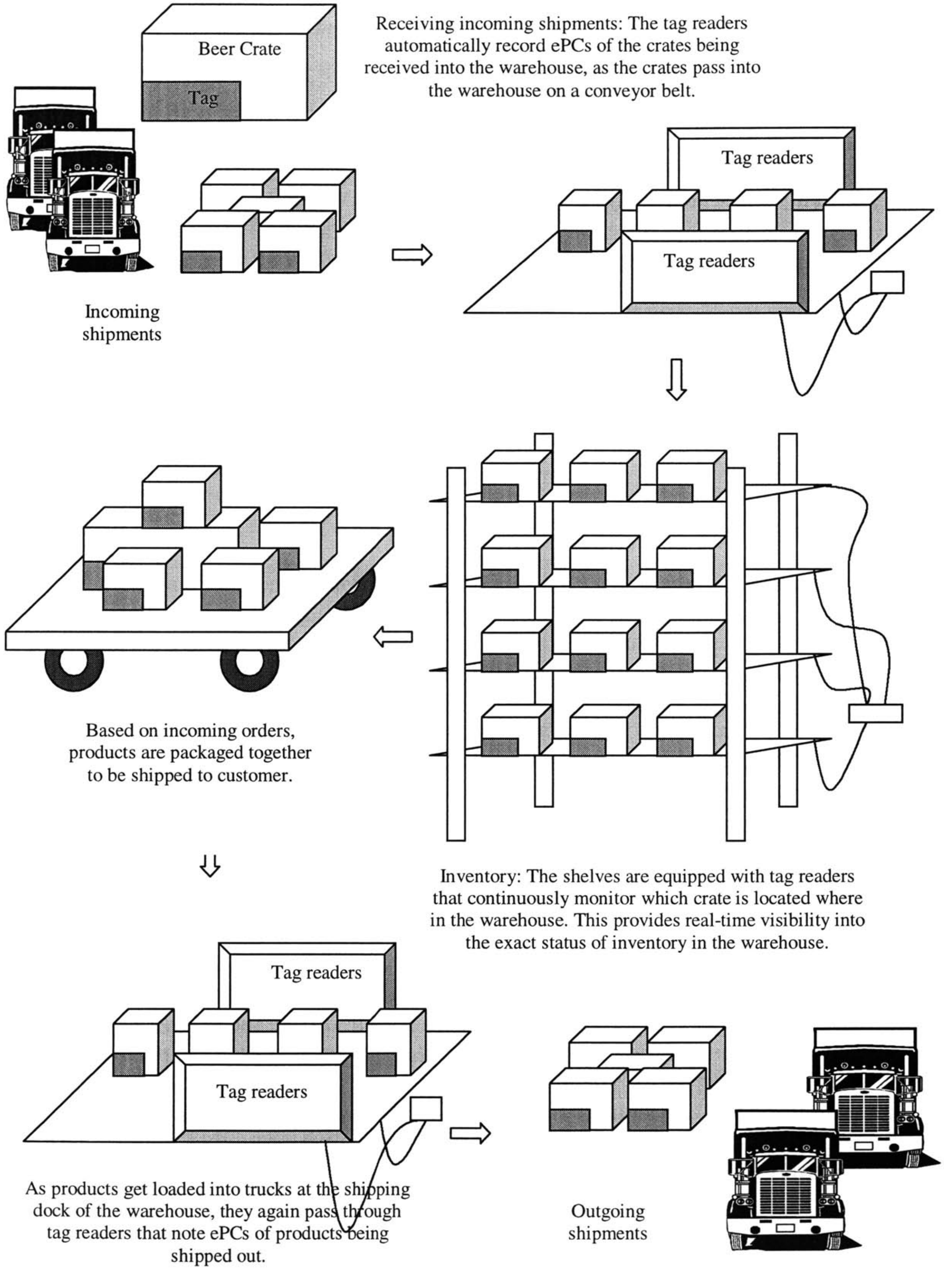


Figure 4.2 Real time data acquisition at the distributor.

4.4.2 Framework extended to the entire supply chain

There is a marked difference in the manner in which information and products flow through traditional supply chains and those of today. In order to achieve faster deliveries, reductions in cost, product customizations, all in all, better customer relationship management; companies today adopt multiple channels of product distribution. This demands information flow in the supply chain to not be serial, but take place from any point to any other point.

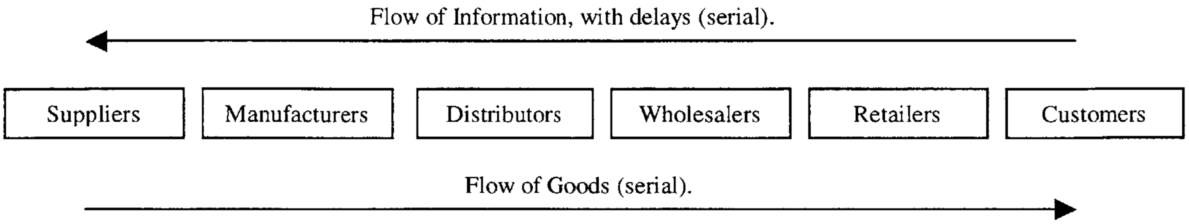


Figure 4.3 Flow in a traditional supply chain.

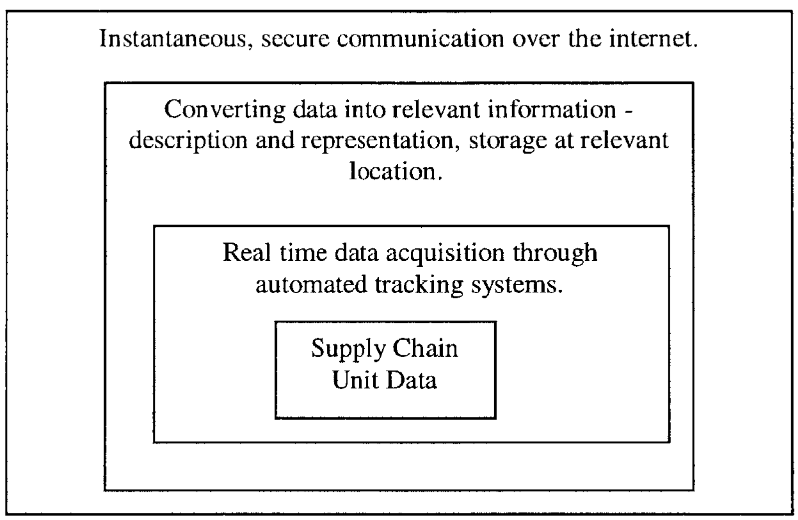


Figure 4.4 Layered approach to information visibility.

In the generic supply chain, we visualize information visibility as shown in Figure 4.4 and 4.5. In each individual supply chain unit, automated tracking systems will acquire data in real time as described above. The data corresponding to objects will be updated and stored at the location described by the ePC. The software application acquiring data in real time will store information pertinent to the unit under consideration in the local unit database. This can contain information about current inventory, the total quantities of different products, individual ePCs of each product, information about upcoming orders and shipping schedules for the unit, etc.The issue of what information should be stored locally is entirely the prerogative of the individual supply chain unit under consideration, and should be resolved with mutual understanding with other partners in the supply chain (since many of them will be involved in contributing to the generation of this

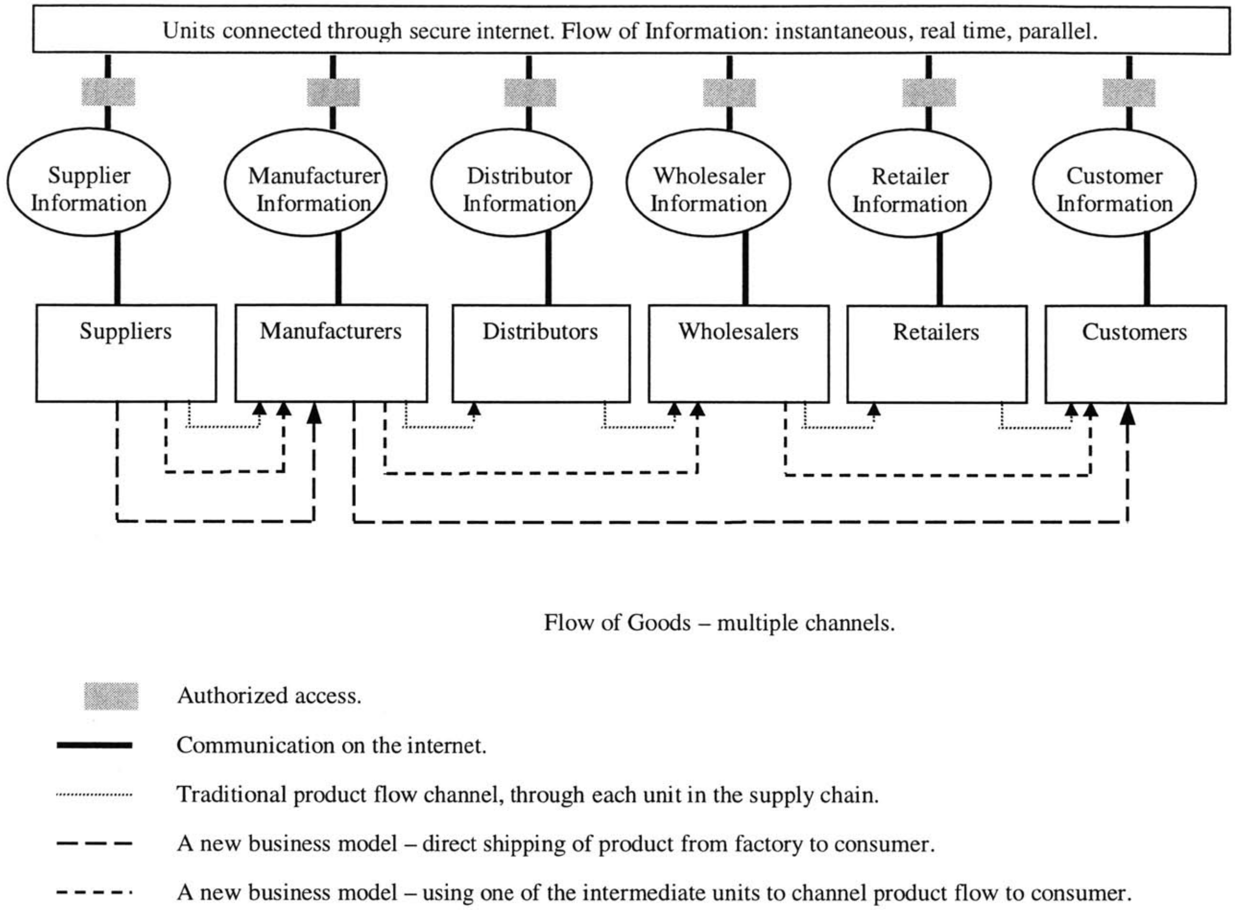


Figure 4.5 Contemporary supply chains.

information). The individual supply chain unit information can be made available on the internet to authorized users. These users can typically be trading partners in the supply chain or end customers. They can use this information for taking further decisions. Provided all trading partners in the product delivery network reach an agreement on implementing such a collaborative framework of data acquisition, description and representation, with instantaneous, secure communication over the internet, we can hope to achieve real time information visibility in the entire product supply chain.

4.5 Implementing the proposed framework in the beer game

The beer distribution network described in the previous chapter consists of four individual units, plus raw materials suppliers to the factory, and end consumers of beer. The information required for determining ordering and shipping quantities and consumer demand forecasting in this supply chain includes past behavior of consumer demand (updated till the latest possible date), inventory levels and backlog levels at each unit.

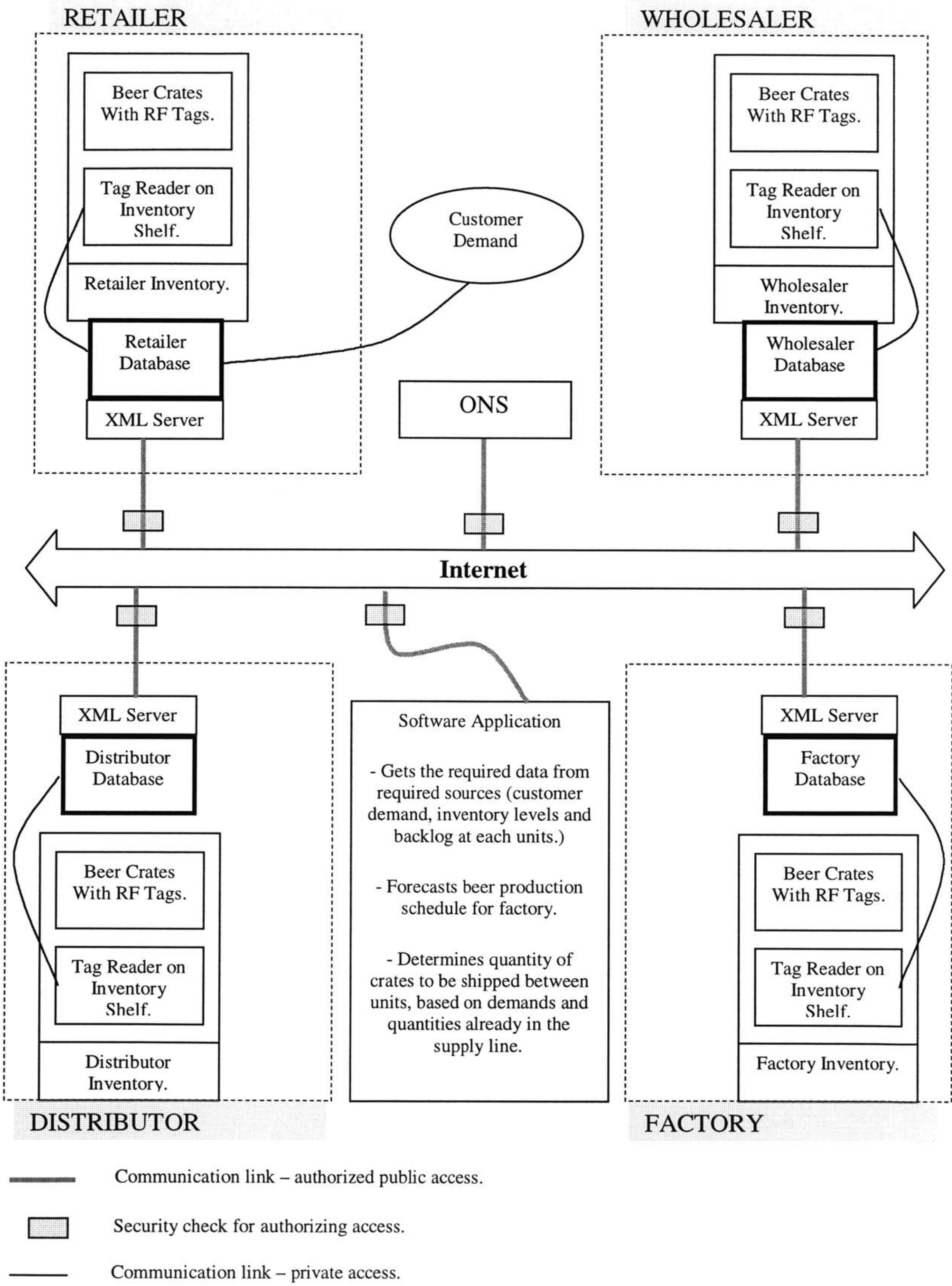


Figure 4.6 Proposed framework for information visibility in the beer game.

We propose that each unit deploy a real time data acquisition system as structured in section 4.4. Since in this example, the most important data required for decision-making in the supply chain is inventory and backlog levels, putting tags on beer crates and equipping inventories in each unit with tag readers is the first step towards achieving information visibility. Tag readers on inventory shelves will read crate ePCs. The software application interfacing with these tag readers will record the ePCs in a local database, as well as lookup the ONS to access the crates description database. It will then update the newly acquired data for the crates under consideration – their new location, tasks completed, next task on hand, etc. The software application can be a locally running program, or a remote program interfacing with tag readers over a network. At each unit, the software application can pass on the data acquired in real time to higher-level enterprise management software. The task of recording weekly customer demand should be assigned to one of the supply chain units – the retailer is in the best position to handle this task. In fact, weekly customer demand can also be extracted from sales per week that get recorded at the retailer.

Chapter 5

Conclusions

The research presented in this thesis is part of a larger initiative to change the way data about physical objects is acquired, understood, represented as useful information, and utilized efficiently. Here, we focus on the data related to products moving in their supply chains.

We began with a review of developments in production and operations management software and supply chain management software over the last three decades. A look into the requirements for successful implementation of these software and requirements of supply chains of today points to real time information visibility as one of the crucial factors for efficient supply chain management. To illustrate impact on costs incurred in the supply chain, we ran computer simulations of supply chain models under differing conditions of information visibility. The results show that information visibility and collaboration provide ~ 40 to 70% reduction in inventory costs alone! Other intangible benefits include reduction in lost sales due to absence of backlogs, improved customer service due to timely delivery of orders, more confidence in managing the supply chain due to accurate, real time knowledge of location of products moving in the supply chain.

We then presented a novel framework for improving information visibility in supply chains. This involves first developing automated object tracking and data acquisition systems – with the help of radio frequency tags and tag readers. These systems eliminate time delays associated with current industrial data acquisition methods, as well as reduce possibility of error in data entry, by minimizing human intervention. The key concept in the proposed data acquisition technique is the use of an electronic product code, ePC, for identifying each single object. This code is currently being developed at MIT's Auto-ID Center. In conjunction with tags and readers, this unique object code provides easy identification and tracking of individual objects. The next step involves converting acquired data into sensible information. The key concepts in the proposed framework for achieving this are the use of an object naming service, ONS, for resolving the location where description about the object is stored, and an object description language, ODL, to describe details about the object being tracked at the latest available time. An interfacing software application does the task of acquiring the ePC from the tag readers, querying thr ONS with the ePC, accessing the location where information about the product is stored, converting recent product specific data acquired into relevant information in an ODL, and updating the location of information storage with the latest available data. Real time information visibility in supply chains is achieved by accessing this data in real time over the internet, from any location in the supply chain.

Crucial to achieving real time information visibility is the widespread adoption of the proposed framework – within a single product supply chain as well as supply chains for different products. Common standards for communication, rules regarding security, authorization and encryption for the use and access of product data are issues that supply chain units will find crucial during implementation of the framework. Under certain situations, trading partners might not wish to reveal their information – safety and privacy of information then becomes crucial to earning the trust of these trading partners.

The rapid development of the internet has led to many changes in ways businesses are run. More and more companies are moving online for procuring raw materials, interacting with customers, and delivering directly to them by bypassing existing channels for product flow. Stores are turning into showrooms; computers are becoming the shopping store; and the shipment delivery boy is replacing the checkout counter. In a world where the customer has a lot of options and loyalty changes with the click of a button, achieving customer satisfaction will take much more than old-fashioned manufacturing shop floors and supply chains. We believe that achieving visibility of information is the first step towards success in this competition.

In conclusion, this thesis outlines an overarching framework. Much needs to be done in terms converting these concepts to reality. The ePC, ONS and an ODL need to be standardized and developed and made viable to be used worldwide as the de-facto standards for object identification and data representation. An industrial experiment, such as implementing a real time data acquisition system in a warehouse (as illustrated in the previous chapter) would be a first step towards providing proof of concept. Further expansion of this system to other units in a supply chain would be the next step. We envision an eventual adoption of this framework in all supply / value chains or trading networks.

References

[1] Quinn F. J., 'The Payoff Potential in Supply Chain Management', *Supply Chain Management Review.*
(http://quinn.ascet.com/)

[2] Cole S.J., Woodring S.D., Chun H., and Gatoff J., 'Dynamic Trading Networks', *The Forrester Report*, January 1999.

[3] McCullough S.S., 'Time to replace your MRP?', *The Forrester Report*, February 1999.

[4] Fogarty D.W., Blackstone J.H., and Hoffman T.R., "Production and Inventory Management", South Western Publishing Company, 1991.

[5] Richards H., 'Notes on Manufacturing Strategy and Performance Measurement Processes', Institute for Manufacturing, University of Cambridge.
(http://www-mmd.eng.cam.ac.uk/people/ahr/dstools/proces/mrp2.htm)

[6] Krajewski L.J., and Ritzman L.P., "Operations Management", Addison-Wesley Publishing Company, 1999.

[7] Kumar K., and Hillegersberg J.V., 'ERP – Experience and Evolution', *Communications of the ACM*, 43(4), pp 22-26, April 2000.

[8] SAP, 'SAP R/3 System Overview'.
(http://www.sap.com/solutions/r3/index.htm)

[9] Stevens G.C., 'Integrating the supply chain', *Intl. J of Physical Distribution and Materials Management*, Vol-19, pp 3-8, 1989.

[10] Tayur S., Ganeshan R., and Magazine M., "Quantitative Models for Supply Chain Management", Kluwer's International Series, 1999.

[11] Hicks D.A., 'A four step methodology for using simulation and optimization technologies in strategic supply chain planning', *Proceedings of the 1999 Winter Simulation Conference*, pp 1215-1220.

[12] Gormley J.H. III, Woodring S.D., and Lieu K.C., 'Supply chain beyond ERP' *The Forrester Report*, May 1997.

[13] Padmos J., Hubbard B., Duczmal T., and Saidi S., 'How i2 integrates simulation in supply chain optimization', *Proceedings of the 1999 Winter Simulation Conference*, pp 1350-1355.

[14] i2 Technologies.
(http://www.i2.com)

[15] Syncra Systems.
(http://www.syncra.com)

[16] Voluntary Interindustry Commerce Standards.
(http://www.vics.org)

[17] Yao A.C., and Carlson J.G., 'The impact of real-time data communication on inventory management', *Intl. J. of Production Economics*, Vol-59, pp 213-219, 1999.

[18] Lindau R., and Lumsden K., 'The use of automatic data capture systems in inventory management', *Intl. J. of Production Economics*, Vol-59, pp 159-167, 1999.

[19] Bruce P., 'Barcode and EDI – "the right stuff" for consumer goods manufacturers', *Automatic I.D. News*, 14(4), pp 46-47, April 1998.

[20] Swamidass P.M., 'Bar code users and their performance', Unova Inc, July 1998.

[21] The AIM Global Network, 'What is RFID?'
(http://www.aimglobal.org)

[22] Loudin A., 'RFID comes into its own', *Warehousing Management*, 6(3), pp 30-32, April 1999.

[23] Motorola, 'Bistatix White paper', Version 4.1, March 1999.
(http://www.motorola.com)

[24] Yating L., and Zhang J., 'Electronic Data Interchange – EDI'.
(http://www.biz.uiowa.edu/class/6k220_park/OldStudProjects/S97/group4/EDI.html)

[25] Erwin B., Callahan P.D., and Cho D., 'Will the web kill EDI?', *The Forrester Report*, February 1996.

[26] Forrester J.W., "Industrial Dynamics", MIT Press, Cambridge, MA, 1961.

[27] Sterman J. D., 'The Beer Distribution Game: An Annotated Bibliography Covering its History and Use in Education and Research', July 1992.
(http://learning.mit.edu/pra/tool/bibl.html)

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

[29] Instructions for playing the beer game.
(http://learning.mit.edu/pra/tool/beer.html)

[30] Kirkwood C. W., 'The Beer Game Production-Distribution Exercise: Running Large Sessions'.
(http://www.public.asu.edu/~kirkwood/sysdyn/BGame/BGame.htm)

[30a] Masters J., 'Lecture Notes – MIT Class on Logistics Systems', Fall 1999.

[31] Sterman J.D., "Business Dynamics: Systems Thinking and Modeling for a Complex World", Irvin McGraw-Hill, 2000.

[33] Kirkwood C. W., 'Business Process Analysis Workshops: System Dynamics Models', Department of Management, Arizona State University, Tempe, AZ 85287-4006.
(http://www.public.asu.edu/~kirkwood/sysdyn/SDWork/SDWork.htm)

[34] Lee H.L., Padmanabhan V., and Whang S., 'The bullwhip effect in supply chains', *Sloan Management Review*, pp 93-102, Spring 1997.

[34a] Hewlett Packard, 'The Bullwhip Effect - Exploring Causes and Counter Strategies'.
(http://www.internetsolutions.enterprise.hp.com/supplychain/library/articles/bullwhip_content.html)

[35] MIT Auto-ID Center - Fact Sheet on ePC.
(http://auto-id.mit.edu/)

[36] MIT Auto-ID Center.
(http://auto-id.mit.edu/)

[36a] Foley J.T., 'An Infrastructure for Electromechanical Appliances on the Internet', S.B. & M.Eng. Thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1999.

[37] MIT Auto-ID Center - Fact Sheet on ONS.
(http://auto-id.mit.edu/)

[38] Intermec News Report, Logistics Management & Distribution Report, June 1999.
(http://www.manufacturing.net/magazine/logistic/archives/1999/log0601.99/491.htm)

[39] TMWG, 'TMWG Recommendations on XML', UN/CEFACT/TMWG/NO89/Rev.5, September 1999.

[40] The World Wide Web Consortium.

(http://www.w3c.org)

[41] Organization for the Advancement of Structured Information Standards.
(http://www.oasis-open.org)

[42] XML.ORG: The XML Industry Portal.
(http://www.xml.org)

[43] ebXML: Electronic Business XML initiative.
(http://www.ebxml.org)

[44] MIT Auto-ID Center - Fact Sheet on PML.
(http://auto-id.mit.edu/)

Appendix A

No:	Variable and defining equation	Comments
(01)	A = 0.26	Forecasting parameter
(02)	B = 0.088	Forecasting parameter
(03)	Backlog = INTEG(bFlow , 0)	Backlog at retailer
(04)	Backlog 0 = INTEG(bFlow 0 , 0)	Backlog at wholesaler
(05)	Backlog 1 = INTEG(bFlow 1 , 0)	Backlog at distributor
(06)	Backlog 2 = INTEG(bFlow 2 , 0)	Backlog at factory
(07)	bFlow = ORDer - sold	Accumulation of backlog at retailer
(08)	bFlow 0 = ordered - sold 0	Accumulation of backlog at wholesaler
(09)	bFlow 1 = ordered 0 - sold 1	Accumulation of backlog at distributor
(10)	bFlow 2 = ordered 1 - sold 2	Accumulation of backlog at factory
(11)	coming = ordered 2	Materials in transit to factory
(12)	Cost = INTEG(cost increase , 0)	Total inventory cost in supply chain
(13)	cost increase = 1 * (Backlog + Backlog 0 + Backlog 1 + Backlog 2) + 0.5 * (Inventory + Inventory 0 + Inventory 1 + Inventory 2)	Weekly inventory cost in supply chain
(14)	Eff Env = Inventory - Backlog	Effective Inventory at retailer
(15)	Eff Inv 0 = Inventory 0 - Backlog 0	Effective Inventory at wholesaler
(16)	Eff Inv 1 = Inventory 1 - Backlog 1	Effective Inventory at distributor
(17)	Eff Inv 2 = Inventory 2 - Backlog 2	Effective Inventory at factory
(18)	FINAL TIME = 36	The final time for the simulation.
(19)	in = DELAY FIXED(sold 0 , 2, 4)	Incoming orders at retailer
(20)	in 0 = DELAY FIXED(sold 1 , 2, 4)	Incoming orders at wholesaler
(21)	in 1 = DELAY FIXED(sold 2 , 2, 4)	Incoming orders at distributor
(22)	in 2 = DELAY FIXED(coming , 2, 4)	Incoming orders at factory
(23)	INITIAL TIME = 0	The initial time for the simulation.
(24)	Inventory = INTEG(in - sold , 12)	Physical inventory at retailer
(25)	Inventory 0 = INTEG(in 0 - sold 0 , 12)	Physical inventory at wholesaler
(26)	Inventory 1 = INTEG(in 1 - sold 1 , 12)	Physical inventory at distributor
(27)	Inventory 2 = INTEG(in 2 - sold 2 , 12)	Physical inventory at factory
(28)	MaxOrder = 100	A limit on the maximum order placed
(29)	ORDer = 4 + STEP (4, 5)	Weekly customer orders
(30)	ordered = DELAY FIXED(placed , 1, 4)	In transit orders by retailer
(31)	ordered 0 = DELAY FIXED(placed 0 , 1, 4)	In transit orders by wholesaler
(32)	ordered 1 = DELAY FIXED(placed 1 , 1, 4)	In transit orders by distributor
(33)	ordered 2 = DELAY FIXED(placed 2 , 1, 4)	In transit orders by factory
(34)	placed = MIN (MAX (0, FORECAST (ORDer , 2, 4) + A * (S - (Inventory - Backlog)) + B * (SL - SupplyL)) , MaxOrder)	Orders placed by retailer
(35)	placed 0 = MIN (MAX (0, FORECAST (ordered , 2, 4) + A * (S - (Inventory 0 - Backlog 0)) + B * (SL - SupplyL 0)) , MaxOrder)	Orders placed by wholesaler
(36)	placed 1 = MIN (MAX (0, FORECAST (ordered 0 , 2, 4) + A * (S - (Inventory 1 - Backlog 1)) + B * (SL - SupplyL 1)) , MaxOrder)	Orders placed by distributor
(37)	placed 2 = MIN (MAX (0, FORECAST (ordered 1 , 2, 4) + A * (S - (Inventory 2 - Backlog 2)) + B * (SL - SupplyL 2)) , MaxOrder)	Orders placed by factory
(38)	S = 12	Forecasting parameter
(39)	SAVEPER = TIME STEP	Frequency at which output is stored.
(40)	sFlow = placed - in	Supply line accumulation - retailer
(41)	sFlow 0 = placed 0 - in 0	Supply line accumulation - wholesaler
(42)	sFlow 1 = placed 1 - in 1	Supply line accumulation - distributor
(43)	sFlow 2 = placed 2 - in 2	Supply line accumulation - factory
(44)	SL = 14.7	Forecasting parameter
(45)	SMOOTHTIME = 1	Forecasting parameter
(46)	sold = MIN (Inventory + in , ORDer + Backlog)	Crates sold by retailer
(47)	sold 0 = MIN (Inventory 0 + in 0 , ordered + Backlog 0)	Crates sold by wholesaler
(48)	sold 1 = MIN (Inventory 1 + in 1 , ordered 0 + Backlog 1)	Crates sold by distributor
(49)	sold 2 = MIN (Inventory 2 + in 2 , ordered 1 + Backlog 2)	Crates sold by factory
(50)	SupplyL = INTEG(sFlow , 8)	Supply line for retailer
(51)	SupplyL 0 = INTEG(sFlow 0 , 8)	Supply line for wholesaler
(52)	SupplyL 1 = INTEG(sFlow 1 , 8)	Supply line for distributor
(53)	SupplyL 2 = INTEG(sFlow 2 , 8)	Supply line for factory
(54)	TIME STEP = 1	The time step for the simulation.

Table A.1 Equations simulating the beer game for Case I.

No:	Variable and defining equation	Comments
(01)	A = 0.26	Forecasting parameter
(02)	B = 0.088	Forecasting parameter
(03)	Backlog = INTEG(bFlow , 0)	Backlog at retailer
(04)	Backlog 0 = INTEG(bFlow 0 , 0)	Backlog at wholesaler
(05)	Backlog 1 = INTEG(bFlow 1 , 0)	Backlog at distributor
(06)	Backlog 2 = INTEG(bFlow 2 , 0)	Backlog at factory
(07)	bFlow = ORDer - sold	Accumulation of backlog at retailer
(08)	bFlow 0 = ordered - sold 0	Accumulation of backlog at wholesaler
(09)	bFlow 1 = ordered 0 - sold 1	Accumulation of backlog at distributor
(10)	bFlow 2 = ordered 1 - sold 2	Accumulation of backlog at factory
(11)	coming = ordered 2	Materials in transit to factory
(12)	Cost = INTEG(cost increase , 0)	Total supply chain cost
(13)	cost increase = 1 * (Backlog + Backlog 0 + Backlog 1 + Backlog 2) + 0.5 * (Inventory + Inventory 0 + Inventory 1 + Inventory 2)	Weekly supply chain cost
(14)	Eff Env = Inventory - Backlog	Effective Inventory at retailer
(15)	Eff Inv 0 = Inventory 0 - Backlog 0	Effective Inventory at wholesaler
(16)	Eff Inv 1 = Inventory 1 - Backlog 1	Effective Inventory at distributor
(17)	Eff Inv 2 = Inventory 2 - Backlog 2	Effective Inventory at factory
(18)	FINAL TIME = 36	The final time for the simulation.
(19)	in = DELAY FIXED(sold 0 , 2, 4)	Incoming orders at retailer
(20)	in 0 = DELAY FIXED(sold 1 , 2, 4)	Incoming orders at wholesaler
(21)	in 1 = DELAY FIXED(sold 2 , 2, 4)	Incoming orders at distributor
(22)	in 2 = DELAY FIXED(coming , 2, 4)	Incoming orders at factory
(23)	INITIAL TIME = 0	The initial time for the simulation.
(24)	Inventory = INTEG(in - sold , 12)	Physical inventory at retailer
(25)	Inventory 0 = INTEG(in 0 - sold 0 , 12)	Physical inventory at wholesaler
(26)	Inventory 1 = INTEG(in 1 - sold 1 , 12)	Physical inventory at distributor
(27)	Inventory 2 = INTEG(in 2 - sold 2 , 12)	Physical inventory at factory
(28)	ORDer = 4 + STEP (4, 5)	Weekly customer orders
(29)	ordered = DELAY FIXED(placed , 1, 4)	In transit orders by retailer
(30)	ordered 0 = DELAY FIXED(placed 0 , 1, 4)	In transit orders by wholesaler
(31)	ordered 1 = DELAY FIXED(placed 1 , 1, 4)	In transit orders by distributor
(32)	ordered 2 = DELAY FIXED(placed 2 , 1, 4)	In transit orders by factory
(33)	placed = MAX (0, SMOOTH (ORDer , SMOOTHTIME) + A * (S - (Inventory – Backlog)) + B * (SL - SupplyL))	Orders placed by retailer
(34)	placed 0 = MAX (0, SMOOTH (ordered , SMOOTHTIME) + A * (S - (Inventory 0 - Backlog 0)) + B * (SL - SupplyL 0))	Orders placed by wholesaler
(35)	placed 1 = MAX (0, SMOOTH (ordered 0 , SMOOTHTIME) + A * (S - (Inventory 1 - Backlog 1)) + B * (SL - SupplyL 1))	Orders placed by distributor
(36)	placed 2 = MAX (0, SMOOTH (ordered 1 , SMOOTHTIME) + A * (S - (Inventory 2 - Backlog 2)) + B * (SL - SupplyL 2))	Orders placed by factory
(37)	S = 12	Forecasting parameter
(38)	SAVEPER = TIME STEP	Frequency at which output is stored.
(39)	sFlow = placed - in	Supply line accumulation - retailer
(40)	sFlow 0 = placed 0 - in 0	Supply line accumulation - wholesaler
(41)	sFlow 1 = placed 1 - in 1	Supply line accumulation - distributor
(42)	sFlow 2 = placed 2 - in 2	Supply line accumulation - factory
(43)	SL = 14.7	Forecasting parameter
(44)	SMOOTHTIME = 1	Forecasting parameter
(45)	sold = MIN (Inventory + in , ORDer + Backlog)	Crates sold by retailer
(46)	sold 0 = MIN (Inventory 0 + in 0 , ordered + Backlog 0)	Crates sold by wholesaler
(47)	sold 1 = MIN (Inventory 1 + in 1 , ordered 0 + Backlog 1)	Crates sold by distributor
(48)	sold 2 = MIN (Inventory 2 + in 2 , ordered 1 + Backlog 2)	Crates sold by factory
(49)	SupplyL = INTEG(sFlow , 8)	Supply line for retailer
(50)	SupplyL 0 = INTEG(sFlow 0 , 8)	Supply line for wholesaler
(51)	SupplyL 1 = INTEG(sFlow 1 , 8)	Supply line for distributor
(52)	SupplyL 2 = INTEG(sFlow 2 , 8)	Supply line for factory
(53)	TIME STEP = 1	The time step for the simulation.

Table A.2 Equations simulating the beer game for Case II.

No:	Variable and defining equation	Comments
(01)	Beer Supply = "F-order"	Raw materials supplied to the beer factory.
(02)	Cost = INTEG(Weekly Costs , 0)	Total Inventory Costs in the supply chain.
(03)	Customer Orders = 4 + STEP (4, 5)	Orders placed by customer.
(04)	"D-Backlog" = INTEG("D-Bflow" , 0)	Backlog at Distributor
(05)	"D-Bflow" = "W-order" - "D-out"	Accumulation of backlog at Distributor.
(06)	"D-in" = DELAY FIXED("F-out" , 2, 4)	Incoming orders at the Distributor.
(07)	"D-order" = MAX (0, FORECAST (Customer Orders , 3, 6) + S - Distributor Inventory + "D-Backlog" + SL - "D-SupplyLine")	Orders placed by the Distributor.
(08)	"D-out" = MIN (Distributor Inventory + "D-in" , "W-order" + "D-Backlog")	Orders shipped out by the Distributor.
(09)	"D-SL-in" = "D-order" - "D-in"	Accumulation in supply line of the Distributor.
(10)	"D-SupplyLine" = INTEG("D-SL-in" , ISL)	Supply line for the Distributor.
(11)	Distributor Inventory = INTEG("D-in" - "D-out" , S)	Inventory Level at the Distributor.
(12)	"F-Backlog" = INTEG("F-Bflow" , 0)	Backlog at Factory.
(13)	"F-Bflow" = "D-order" - "F-out"	Accumulation of backlog at Factory.
(14)	"F-in" = DELAY FIXED(Beer Supply , 2, 4)	Incoming orders at the Factory.
(15)	"F-order" = MAX (0, FORECAST (Customer Orders , 4, 8) + S - Factory Inventory + "F-Backlog" + SL - "F-SupplyLine")	Orders placed by the Factory.
(16)	"F-out" = MIN (Factory Inventory + "F-in" , "D-order" + "F-Backlog")	Orders shipped out by the Factory.
(17)	"F-SL-in" = "F-order" - "F-in"	Accumulation in supply line of the Factory.
(18)	"F-SupplyLine" = INTEG("F-SL-in" , ISL)	Supply line for the Factory.
(19)	Factory Inventory = INTEG("F-in" - "F-out" , S)	Inventory Level at the Factory.
(20)	FINAL TIME = 36	The final time for the simulation.
(21)	INITIAL TIME = 0	The initial time for the simulation.
(22)	ISL = 8	Initial Supply line level.
(23)	"R-Backlog" = INTEG("R-Bflow" , 0)	Backlog at Retailer.
(24)	"R-Bflow" = Customer Orders - "R-out"	Accumulation of backlog at Retailer.
(25)	"R-in" = DELAY FIXED("W-out" , 2, 4)	Incoming orders at the Retailer.
(26)	"R-order" = MAX (0, FORECAST (Customer Orders , 1, 2) + S - Retailer Inventory + "R-Backlog" + SL - "R-SupplyLine")	Orders placed by the Retailer.
(27)	"R-out" = MIN (Retailer Inventory + "R-in" , Customer Orders + "R-Backlog")	Orders shipped out by the Retailer.
(28)	"R-SL-in" = "R-order" - "R-in"	Accumulation in supply line of the Retailer.
(29)	"R-SupplyLine" = INTEG("R-SL-in" , ISL)	Supply line for the Retailer.
(30)	Retailer Inventory = INTEG("R-in" - "R-out" , S)	Inventory Level at the Retailer.
(31)	S = 12	Desired inventory level.
(32)	SAVEPER = TIME STEP	The frequency with which output is stored.
(33)	SL = 16	Desired supply line level.
(34)	TIME STEP = 1	The time step for the simulation.
(35)	"W-Backlog" = INTEG("W-Bflow" , 0)	Backlog at Wholesaler.
(36)	"W-Bflow" = "R-order" - "W-out"	Accumulation of backlog at Wholesaler.
(37)	"W-in" = DELAY FIXED("D-out" , 2, 4)	Incoming orders at the Wholesaler.
(38)	"W-order" = MAX (0, FORECAST (Customer Orders , 2, 4) + S - Wholesaler Inventory + "W-Backlog" + SL - "W-SupplyLine")	Orders placed by the Wholesaler.
(39)	"W-out" = MIN (Wholesaler Inventory + "W-in" , "R-order" + "W-Backlog")	Orders shipped out by the Wholesaler.
(40)	"W-SL-in" = "W-order" - "W-in"	Accumulation in supply line of the Wholesaler.
(41)	"W-SupplyLine" = INTEG("W-SL-in" , ISL)	Supply line for the Wholesaler.
(42)	Weekly Costs = 0.5 * (Retailer Inventory + Wholesaler Inventory + Distributor Inventory + Factory Inventory) + 1 * ("R-Backlog" + "W-Backlog" + "D-Backlog" + "F-Backlog")	Weekly Inventory Costs in the supply chain.
(43)	Wholesaler Inventory = INTEG("W-in" - "W-out" , S)	Inventory Level at the Wholesaler.

Table A.3 Equations simulating the beer game for Case III.

No:	Variable and defining equation	Comments
(01)	Beer Supply = "F-order"	Raw materials supplied to the beer factory.
(02)	Cost = INTEG (Weekly Costs , 0)	Total Inventory Costs in the supply chain.
(03)	Customer Orders = 4 + STEP (4, 5)	Orders placed by customer.
(04)	"D-Backlog" = INTEG("D-Bflow" , 0)	Backlog at Distributor
(05)	"D-Bflow" = "W-order" - "D-out"	Accumulation of backlog at Distributor.
(06)	"D-in" = DELAY FIXED("F-out" , 2, 4)	Incoming orders at the Distributor.
(07)	"D-order" = MAX (0, SMOOTH (Customer Orders , 1) + S - Distributor Inventory + "D-Backlog" + SL - "D-SupplyLine")	Orders placed by the Distributor.
(08)	"D-out" = MIN (Distributor Inventory + "D-in" , "W-order" + "D-Backlog")	Orders shipped out by the Distributor.
(09)	"D-SL-in" = "D-order" - "D-in"	Accumulation in supply line of the Distributor.
(10)	"D-SupplyLine" = INTEG("D-SL-in" , ISL)	Supply line for the Distributor.
(11)	Distributor Inventory = INTEG("D-in" - "D-out" , S)	Inventory Level at the Distributor.
(12)	"F-Backlog" = INTEG("F-Bflow" , 0)	Backlog at Factory.
(13)	"F-Bflow" = "D-order" - "F-out"	Accumulation of backlog at Factory.
(14)	"F-in" = DELAY FIXED(Beer Supply , 2, 4)	Incoming orders at the Factory.
(15)	"F-order" = MAX (0, SMOOTH (Customer Orders , 1) + S - Factory Inventory + "F-Backlog" + SL - "F-SupplyLine")	Orders placed by the Factory.
(16)	"F-out" = MIN (Factory Inventory + "F-in" , "D-order" + "F-Backlog")	Orders shipped out by the Factory.
(17)	"F-SL-in" = "F-order" - "F-in"	Accumulation in supply line of the Factory.
(18)	"F-SupplyLine" = INTEG("F-SL-in" , ISL)	Supply line for the Factory.
(19)	Factory Inventory = INTEG("F-in" - "F-out" , S)	Inventory Level at the Factory.
(20)	FINAL TIME = 36	The final time for the simulation.
(21)	INITIAL TIME = 0	The initial time for the simulation.
(22)	ISL = 8	Initial Supply line level.
(23)	"R-Backlog" = INTEG("R-Bflow" , 0)	Backlog at Retailer.
(24)	"R-Bflow" = Customer Orders - "R-out"	Accumulation of backlog at Retailer.
(25)	"R-in" = DELAY FIXED("W-out" , 2, 4)	Incoming orders at the Retailer.
(26)	"R-order" = MAX (0, SMOOTH (Customer Orders , 1) + S - Retailer Inventory + "R-Backlog" + SL - "R-SupplyLine")	Orders placed by the Retailer.
(27)	"R-out" = MIN (Retailer Inventory + "R-in" , Customer Orders + "R-Backlog")	Orders shipped out by the Retailer.
(28)	"R-SL-in" = "R-order" - "R-in"	Accumulation in supply line of the Retailer.
(29)	"R-SupplyLine" = INTEG("R-SL-in" , ISL)	Supply line for the Retailer.
(30)	Retailer Inventory = INTEG("R-in" - "R-out" , S)	Inventory Level at the Retailer.
(31)	S = 12	Desired inventory level.
(32)	SAVEPER = TIME STEP	The frequency with which output is stored.
(33)	SL = 16	Desired supply line level.
(34)	TIME STEP = 1	The time step for the simulation.
(35)	"W-Backlog" = INTEG("W-Bflow" , 0)	Backlog at Wholesaler.
(36)	"W-Bflow" = "R-order" - "W-out"	Accumulation of backlog at Wholesaler.
(37)	"W-in" = DELAY FIXED("D-out" , 2, 4)	Incoming orders at the Wholesaler.
(38)	"W-order" = MAX (0, SMOOTH (Customer Orders , 1) + S - Wholesaler Inventory + "W-Backlog" + SL - "W-SupplyLine")	Orders placed by the Wholesaler.
(39)	"W-out" = MIN (Wholesaler Inventory + "W-in" , "R-order" + "W-Backlog")	Orders shipped out by the Wholesaler.
(40)	"W-SL-in" = "W-order" - "W-in"	Accumulation in supply line of the Wholesaler.
(41)	"W-SupplyLine" = INTEG("W-SL-in" , ISL)	Supply line for the Wholesaler.
(42)	Weekly Costs = 0.5 * (Retailer Inventory + Wholesaler Inventory + Distributor Inventory + Factory Inventory) + 1 * ("R-Backlog" + "W-Backlog" + "D-Backlog" + "F-Backlog")	Weekly Inventory Costs in the supply chain.
(43)	Wholesaler Inventory = INTEG("W-in" - "W-out" , S)	Inventory Level at the Wholesaler.

Table A.4 Equations simulating the beer game for Case IV.