System Dynamics Analysis of an Ordering System used for Commercial Aircraft Manufacture

by
Jeffrey John Finan

B.S. Astronautical Engineering, USAF Academy (1984)

Submitted to the Department of Management and
the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Management
and
Master of Science in Aeronautics and Astronautics

in Conjunction with the Leaders for Manufacturing Program
at the
Massachusetts Institute of Technology
May 1993

Signature of Author____________________________
Departments of Management and Aeronautics and Astronautics
May 14, 1993

Certified by____________________________
John D. Sterman
Associate Professor, Management Science
Thesis Supervisor

Certified by____________________________
Eugene E. Covert
T. Wilson Professor of Aeronautics
Thesis Supervisor

Accepted by____________________________
Jeffrey A. Barks
Associate Dean, Sloan Master's and Bachelor's Programs

Accepted by____________________________
Professor Harold Y. Wachman
Chairman, Department of Aeronautics & Astronautics Graduate Committee

SYSTEM DYNAMICS ANALYSIS OF AN ORDERING SYSTEM USED FOR COMMERCIAL AIRCRAFT MANUFACTURE

BY

Jeffrey John Finan

Submitted to the Sloan School of Management and the Department of Aeronautics & Astronautics in Partial Fulfillment of the Requirements for the Degrees of Master of Science in Management and Master of Science in Aeronautics and Astronautics

ABSTRACT

The Boeing Company has had an impressive history in building commercial aircraft. In the last few years, however, many factors have combined to create record losses for domestic air carriers. As a result, Boeing has come under severe pressure to reduce the cost of aircraft manufacture. To a large extent, Boeing produces aircraft using the "mass production" paradigm. Inherent within this manufacturing system are large lot sizes and long process cycle times. Despite the high level of inventory used within Boeing work centers, component shortages are frequent. As Boeing becomes a lean producer, it will have to make adjustments in how it orders parts to support the new manufacturing environment.

In this thesis, I study the dynamic behavior of the Boeing ordering system to determine the system's contribution to excess inventory and component shortages. System Dynamics modeling is the tool used to perform the analysis. The basic assumption of this method is that the complex behavior often seen within systems is a result of their causal structure. Precise system optimization is not the goal of this technique, but rather the goal is to increase understanding of the system and identify changes that will provide the greatest leverage for system improvement.

I conclude that the following items have the greatest potential for improving system performance: 1) consistent ordering policies across component types, 2) order synchronization between assemblies and detail components, 3) use of simple safety stock levels, 4) a new business system architecture, 5) reduction in supplier reorder lead times, 6) use of lean production techniques, 7) more accurate metrics, and 8) a systems approach to process improvement.

Thesis supervisors:
Eugene E. Covert, T. Wilson Professor of Aeronautics
John Sterman, Associate Professor of Management Science

ACKNOWLEDGMENTS

I gratefully acknowledge the support and resources made available to me through the Leaders for Manufacturing (LFM) Program, a partnership between MIT and thirteen major U.S. manufacturing companies. Especially to the many people at Boeing who helped me with their time and insights, thank you for being so supportive during my internship. In particular, I would like to extend special thanks to David Fitzpatrick and Frank Hughes.

To my thesis advisors, John Sterman and Gene Covert, thank you for your patience and support. To my LFM classmates Lindsay Anderson, Kevin Bartelson, and Mark Lundstrom, thank you for creating a creative and fun internship experience.

To my family and friends, without whom I would not have made it through the difficult times, thank you for being the light that guided me through the stormy seas to safe harbor.

And to Miriam, whose memory reminds me of all the good things in life, I miss you. You allowed me to love and be loved and taught me a new measure of success that will stay with me always.

> *To Laugh often and much; to win the respect of intelligent people and earn affection of children; to earn the appreciation of honest critics and endure the betrayal of false friends; to appreciate beauty; to find the best in others; to leave the world a bit better; to know even one life has breathed easier because you have lived. This is to have succeeded.*
>
> *- Ralph Waldo Emerson -*

TABLE OF CONTENTS

APPENDICES

Appendix A - System Dynamics Model

Appendix B - Data

Table of Figures

Table of Tables

Chapter 1 - Introduction

The current market environment for airframe manufacturers does not look promising. Downsizing is under way at most of the major players within the industry. Boeing has announced work force reductions totaling 20,000 employees over the next two years.[1] Pratt & Whitney, a major engine manufacturer, has announced similar reductions of 10,000.[2] A variety of factors such as global economic stagnation, over expansion of routes, over capacity, and fare wars have combined to create over $10 billion in losses for domestic carriers over the last three years as seen in Fig. 1.1 . [3]

Price wars from one level on the value chain often work their way back to the suppliers. Over the past several years, airframe manufacturers have had to use "creative financing" to secure new business. As long as the financial condition of most of the major airlines remains weak, the airframe manufacturer who is the lowest cost and most efficient producer will have an advantage.

The MIT Commission on Industrial Productivity saw this trend when it conducted a review of the commercial aircraft industry in the late 1980s. The Made In America study forecasted an environment where "cost-effective management of development and operations" would be necessary for airframe manufacturers to continue to successfully compete for customers.

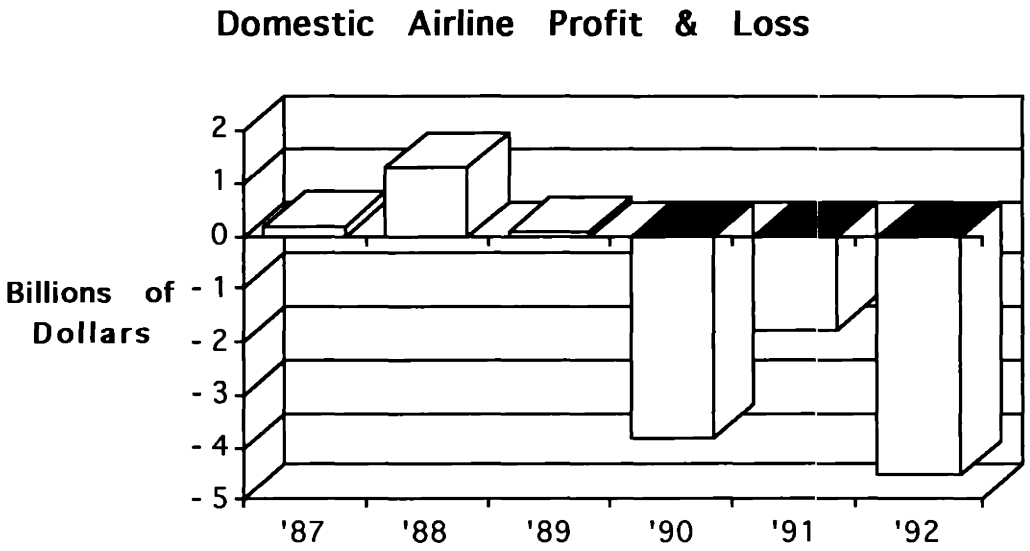
---

[1] From Reuters News Service, "Boeing lays off 224 workers", March 20, 1993.

[2] From The Hartford Courant, "Pratt may cut more jobs", March 19, 1993.

[3] From The New York Times, "Clinton considers measures to help troubled airlines", March 22, 1993, p. A1.

Boeing senior management recognized this as well back in 1986. Several Boeing initiatives have been undertaken to improve productivity and reduce costs as seen in Table 1.1 below.



Source: Government Accounting Office (GAO)

Figure 1.1 - Domestic Airline Profit & Loss

Table 1.1 - Boeing Initiatives to Improve Productivity & Reduce Costs

Continuous Quality Improvement (CQI)

Work Management

Hardware Variability Control (HVC)

Advanced Quality System (AQS)

Flowtime Reduction

Boeing's success with these efforts continues to gain momentum, but the overall manufacturing structure that Boeing uses is still based on the mass production paradigm. Based on this manufacturing approach, there are limits to the productivity improvements that can be realized. The data shown in Table 1.2 provides a comparison of performance results between mass and lean producers. This data, from the MIT International Motor Vehicle Program (IMVP), provides evidence of the benefits of the lean manufacturing approach.

Table 1.2 - Performance Comparison of Lean vs. Mass Producer

Characteristic	**Mass Producer**	**Lean Producer**
Assembly Hours per Car	31	16
Assembly Defects per 100 Cars	135	45
Assembly Space per Car	8.1	4.8
Inventories of Parts (average)	2 weeks	2 hours

Source: IMVP World Assembly Plant Survey

What exactly does it mean to be a lean manufacturer ? There are many different answers to this question, but for the purposes of this discussion, lean manufacturers are driven by three primary factors: 1) inventory reduction, 2) cycle time reduction, and 3) a product focus. To implement this production philosophy, the design of the overall manufacturing system is of critical importance.

Within Boeing, the manufacturing system is composed of hundreds of different production islands that require redundant infrastructure for integration. One Boeing executive, Mr. David Fitzpatrick, believes that the current organization of manufacturing elements leads to excessive "linkage costs". An example of a "linkage cost" would be multiple storage areas for components as they move through the production process. Each time the component is stored, it is inspected, cataloged, put in storage, and later retrieved from storage. All of these activities are termed "non-value added" because the product is not improved by these actions. Lean manufacturers move aggressively to eliminate "linkage costs".

Boeing has been slow to alter its production system. The inertia of the current manufacturing approach presents a strong obstacle to change. It is also never easy to make changes within manufacturing while there are continuing production requirements. Improving operations takes a back seat to moving product out the door. It is my hope that this thesis can provide one tool that will help Boeing managers make the transition to a lean production system.

Focus of the Study

In this thesis, the dynamic behavior of an ordering system used for commercial aircraft manufacture is analyzed to determine the system's contribution to excess inventory and component shortages. Recommendations are made regarding policy and system structural changes that may reduce these problems. In addition, structural changes within the production environment that will remove non-value added linkages and improve system responsiveness are addressed.

Background

The design and manufacture of commercial aircraft is a complex undertaking. For example, the Boeing 747 aircraft is composed of 161,000 unique part numbers and over 1,153,800 actual parts, not including all of the common fasteners such as rivets. In many industries, problems with ordering and tracking parts that make up the product are commonplace. Because of variation in schedules for engineering design, manufacturing process development, actual component fabrication, etc. it is extremely difficult to make sure that parts are available when they are required. This is true in industries where the total product part count is a fraction of that required to build a commercial aircraft; the problem is magnified 1000 fold for airframe manufacturers.

To reduce part shortages, many different approaches have been used. Ordering policy has been designed to use inventory as a buffer to prevent frequent shortages. Organizational structures have been developed to track parts from origin to destination because of system deficiencies, and expediters have been needed to move parts quickly through the production process. All of these systems are used to improve the chances of component availability, but at great cost.

Despite the emphasis on reducing shortages, the manufacturing environment continues to suffer from a relatively high level of variation in production schedules directly attributable to part shortages. In addition, Boeing recognizes that its high level of inventory, large organizational infrastructure, and complex business systems make it more difficult to remain cost competitive within the industry.

Relevant Research

The available research on this topic, as applied to commercial aircraft manufacturing, is rather limited. General evaluations of the dynamic behavior of production-distribution systems have been performed by Forrester [1961], Lyneis [1980], and Morecroft [1983]. Their research highlights the basic structure of ordering systems but differs in significant aspects from this research. Demand within the commercial aircraft industry, due to the long lead times, is fairly deterministic. The producer develops a master schedule in which production rates change infrequently. This differs with the work of Forrester and Lyneis where demand is considered to be stochastic. The specific policies and contingency responses of Boeing's ordering system also differ from those used in prior research and leads to a unique dynamic response.

Prior research by Whiting [1992] and Copes [1992], Leaders For Manufacturing Fellows in the Class of 1992, proved to be beneficial. Whiting looked at order processing within the Sheet Metal Center (SMC) at Boeing. Her thesis evaluated the impact of the order release process on factory delivery performance. Specifically, she concluded that 1) the pre-release process in the SMC was controlled by the corporate upstream ordering process, 2) reduction in the variation of the lead time distribution could improve SMC flowtime performance, and 3) a need exists for greater coordination between the SMC and support agencies during the pre-release process. Whiting's research provided general insights into the problems faced by the production center as a result of ordering system behavior, but the research was limited to the order release process at the SMC and did not analyze the ordering process from the source of requirements to the production centers.

Copes identified factors within the internal production environment at the SMC that determine flowtime performance. Cope's research helped to provide understanding of the characteristics and behavior of the SMC production system which was useful for structural representation of the model developed in this thesis.

Thesis Overview

Chapter 2 - An Ordering System Used for Commercial Aircraft Manufacture

Boeing's manufacturing architecture is briefly introduced, followed by a more involved discussion of how parts are ordered within that structure. Current problems within the ordering system and their causes are reviewed.

Chapter 3 - System Dynamics Analysis of the Boeing Ordering System

The method used for system dynamics analysis of the Boeing ordering system is introduced. The specific model developed is discussed along with key assumptions.

Chapter 4 - Results of Analysis

The results of this analysis are presented and evaluated. The validity of the model is established by comparing model output with reference behavior seen within the current ordering system.

Chapter 5 - Policy Analysis

The search for leverage points within the system is discussed. Different policy approaches are attempted to find potential sources of system improvement and degradation.

Chapter 6 - Conclusions and Recommendations

A summary of conclusions and recommendations is presented along with possible areas for further research.

Chapter 2 - An Ordering System Used For Commercial Aircraft Manufacture

Structural Relationships in the Production Environment

Three major elements contribute to the manufacture of commercial aircraft for the Boeing Company as shown in Figure 2.1: the Boeing Commercial Airplane Group (BCAG), internal suppliers, and external suppliers. The approximate percentage of cost dollars attributable to internal suppliers, external suppliers, and engines can be seen in Figure 2.2. [1]

BCAG facilities are located in the state of Washington and Portland, Oregon. Examples of the types of products manufactured within Component Fabrication are listed in Table 2.1. The complexity and cost of these components varies significantly. Simple sheet metal parts which cost a few dollars to produce are manufactured as are wing panels which cost thousands of dollars to produce. Raw materials required to fabricate these components are ordered from external suppliers by Boeing's Materiel Division. Depending on the intended use of the fabricated component, it will either be delivered to the Component Assembly areas or to a Major Assembly area.

For example, a small sheet metal part used as a structural member of a subassembly will be delivered to a Component Assembly area. Eventually that subassembly will be delivered to a Major Assembly area where it will be installed into the aircraft. A wing spar, on the other hand, will be delivered right to a Major Assembly area.

---

[1] 1989 Inventory Management Organization data.

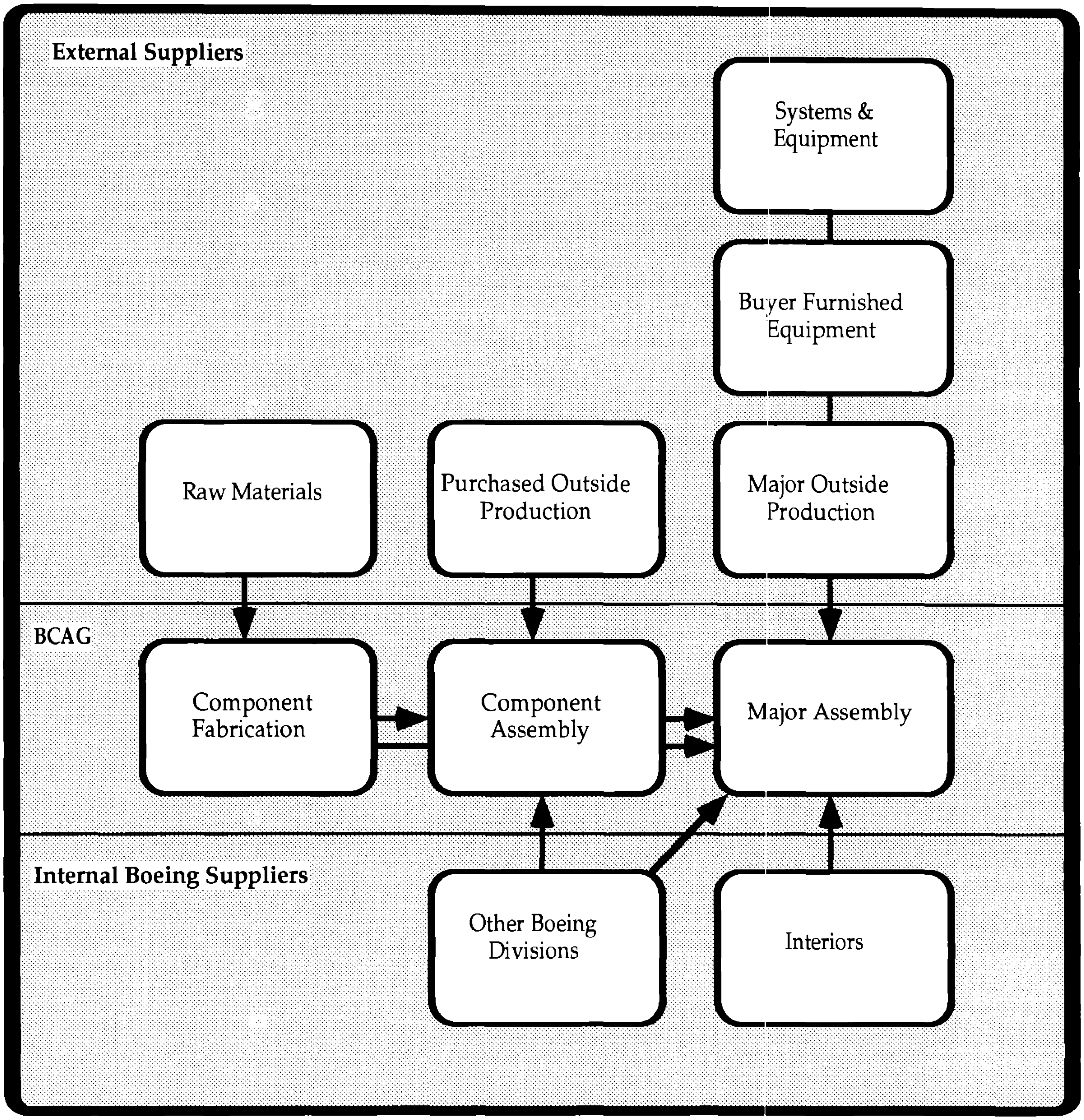


Fig. 2.1 - Boeing Manufacturing Organizational Structure

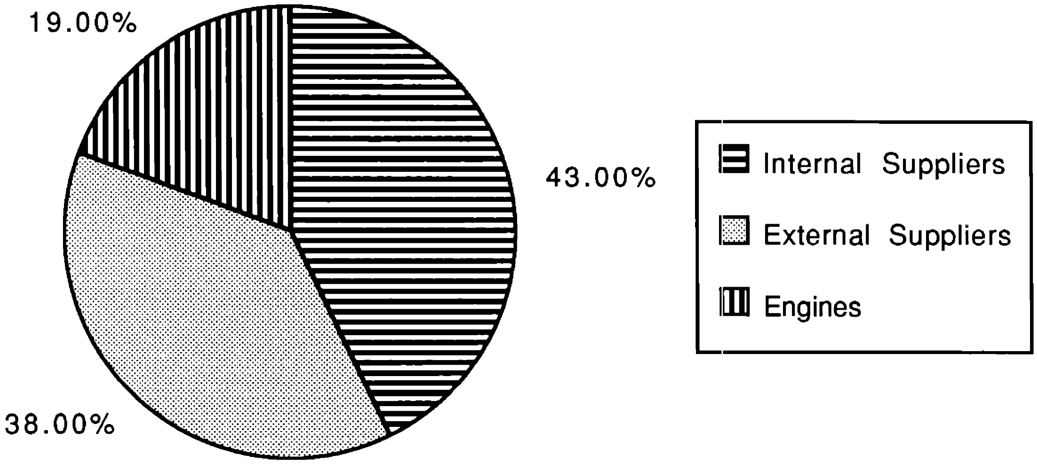


Fig. 2.2 - Allocation of Production Costs (1989 data)

Table 2.1 - Components Manufactured at Fabrication Division

- Wing Skins & Spars
- Sheet Metal Parts
- Floor Panels
- Composite Parts
- Welded Ducts
- Machined Parts
- Landing Gear Beams
- Tooling

Component Assembly uses components as building blocks to construct larger assemblies which will eventually be used in final assembly. Detail components (the individual components that make up an assembly) arrive from several different sources. Some details come from Component Fabrication, as previously mentioned. Others come from Purchased Outside Production (POP), where sheet metal and machined parts are procured from external suppliers. In addition, some details are produced by other Boeing Divisions: Wichita, Georgia, Winnipeg, Vertol, ESD, Oakridge, Boeing Aerospace, or Arnprior.

Major Assembly consists of the final assembly of all aircraft sections. Aircraft sections travel along an assembly line moving from one area to the next by using large overhead cranes. Eventually the landing gear are attached, and the aircraft rolls forward in the assembly line. Major Assembly is the final destination for all components provided by internal or external suppliers.

Major Outside Production, which consists of large, very expensive components, feeds Major Assembly. Once an aircraft program is launched, Boeing selects a few primary suppliers. These suppliers are responsible for large components used on the aircraft. For example, on the 767 program a Japanese supplier provides body panels used in the fuselage. Other large components like the empennage or aircraft control surfaces are also frequently part of the Major Outside Production work packages.

Buyer Furnished Equipment consists of items such as galleys and lavatories. The customer orders these components and provides them for use in Major

Assembly. This equipment is delivered to the final assembly line where it is installed in the designated aircraft.

Systems & Equipment consists of a variety of different high value items. Avionics and instrumentation systems are examples of the types of items procured.

Boeing produces most of the interior elements of its aircraft. In the later stages of Major Assembly, the aircraft interiors arrive in trucks and are installed in the aircraft.

Process Flow for Creating an Order

Due to the size and cost of some of the components discussed above, they are manually ordered and tracked. For example, if a Major Outside Production shipment of body panels are due to arrive from a Japanese supplier, this shipment will have a specific schedule associated with it, and it will be tracked to that schedule. Many of the thousands of other details and assemblies used in constructing an aircraft are ordered and tracked by a network of business systems. These business systems and the parts that they order and track are the primary focus of this research.

As identified in Figure 2.3, several steps are required to generate an order for a commercial aircraft component. First, design engineering for the component must be complete. The engineering drawing identifies the characteristics of the part and its effectivity. The effectivity of the part will specify how many aircraft will need this particular configuration.

Manufacturing Engineering receives the drawing and develops the manufacturing plans which describe how to build the part. The manufacturing plan also contains the estimated setup and runs times of all process steps.

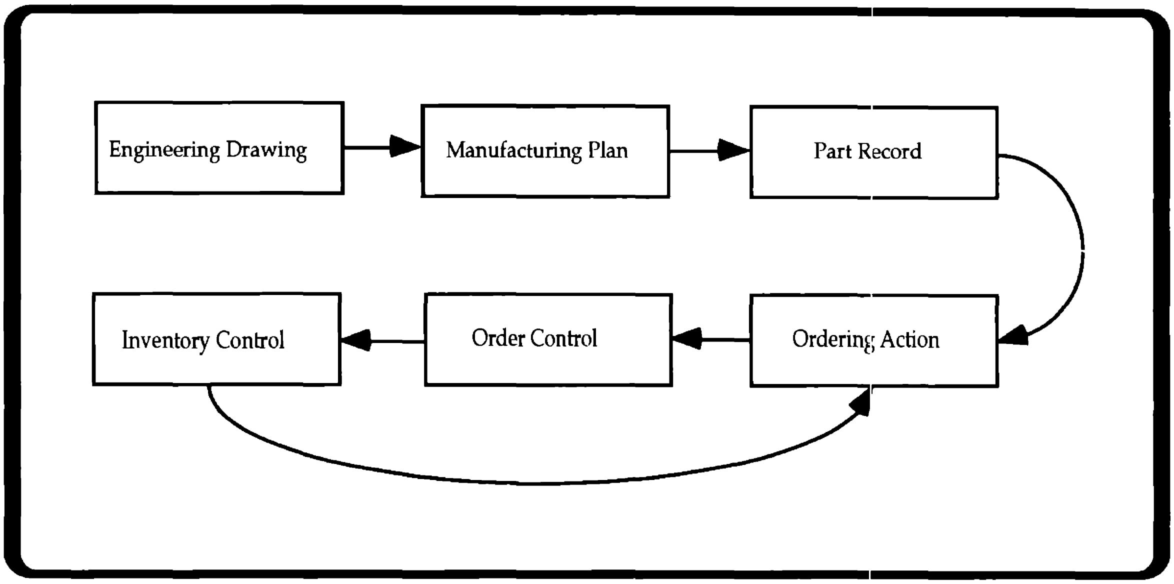


Fig. 2.3 - Process Flow of Ordering System

Manufacturing engineering also builds a part record for the component which will be used to order the part from the appropriate source. The following items can be found on the part record and are used throughout the ordering process:

procurement code - identifies the source of procurement;
effectivity - identifies the production requirements;
parent ID - identifies the assembly that the detail is used on;
flow time allocation - time allocated for part fabrication & transfer;
increment code - the cost category of the part defined by its unit cost.

The Inventory Management Organization (IMO) reviews the part record for accuracy. After validation, the Scheduled Requirements & Order Analysis (SROA) system, where ordering action takes place, receives the part record. SROA determines the point in the future when the current supply of components will be depleted and generates a new order. SROA determines the new order release date by subtracting the appropriate flowtime from the projected non-availability date of the component in question. For example, if inventory, for an assembly with a thirty day flowtime, is scheduled to be depleted, and additional inventory will not be available for assembly load on manufacturing day (M-day) 10,030; the next order, which will replenish the inventory level, will start processing on M-day 10,000 (Non-availability date - planned flowtime = 10,030 - 30 = 10,000).

The ordering policy embedded within the SROA system controls the order quantity. For internally produced components, details are ordered based on Economic Order Quantity (EOQ) logic, and assemblies are ordered based on Time Span ordering logic. For externally supplied parts, Time Span ordering logic is used.

The EOQ formulation reflects the tradeoff between the procurement cost and the inventory holding cost. For a detailed example of EOQ ordering policy and a discussion of many of the problems with this approach, please refer to Appendix B-1. Time Span ordering builds lots based on production requirements over a given period of time. For example, if the current use rate for an assembly is 3 per day, and the plan is to produce in lot sizes based on a 10 day time span; the lot size used for this part would be 30 units (3 units per day * 10 day time span = 30 units).

Protective ordering is used for some components. With protective ordering, excess units are ordered in addition to normal production requirements to counteract losses within the production system due to product quality problems or other unanticipated events. When protective ordering is utilized it leads to order sizes up to 1.5 times the normal amount.

Puget Sound Flow[2] dictates the flowtime available for internal suppliers, and it allocates fixed amounts of time for various activities within the manufacturing flow. Figure 2.4 below shows the various elements of one type of standard flow. Part fabrication flowtime varies depending on the complexity of the part. Flowtime Between Operations (FTBO) eats up twenty days of flowtime. FTBO provides time to accomplish transfer of the part, storage in the proper bin location, and other administrative action prior to the required on-dock date for use by the next operation. In this example, the Lot Time area takes this detail and others and builds a more complex

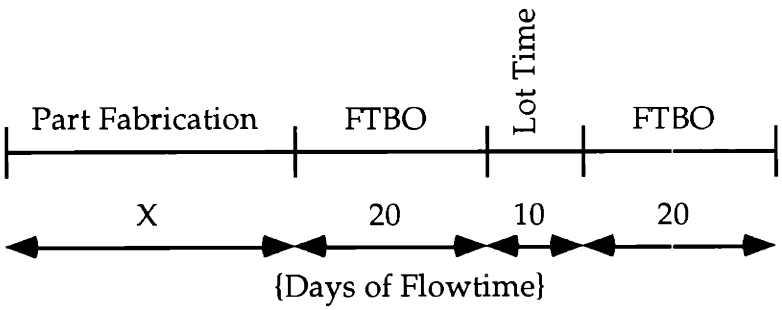


Fig. 2.4 - Elements of Puget Sound Flow

---

[2] Puget Sound Flow Standards are identified in Boeing Document M12-02; this document attempts to identify the normal time required for "accomplishment of all manufacturing tasks required to complete a planned sequence of operations from the initiation of an order to the master scheduled use date of an item".

component. Lot Time assemblies always receive ten days of flowtime no matter how trivial or complex the component, and Lot Time is followed by another twenty days of FTBO prior to use in a Major Assembly area.

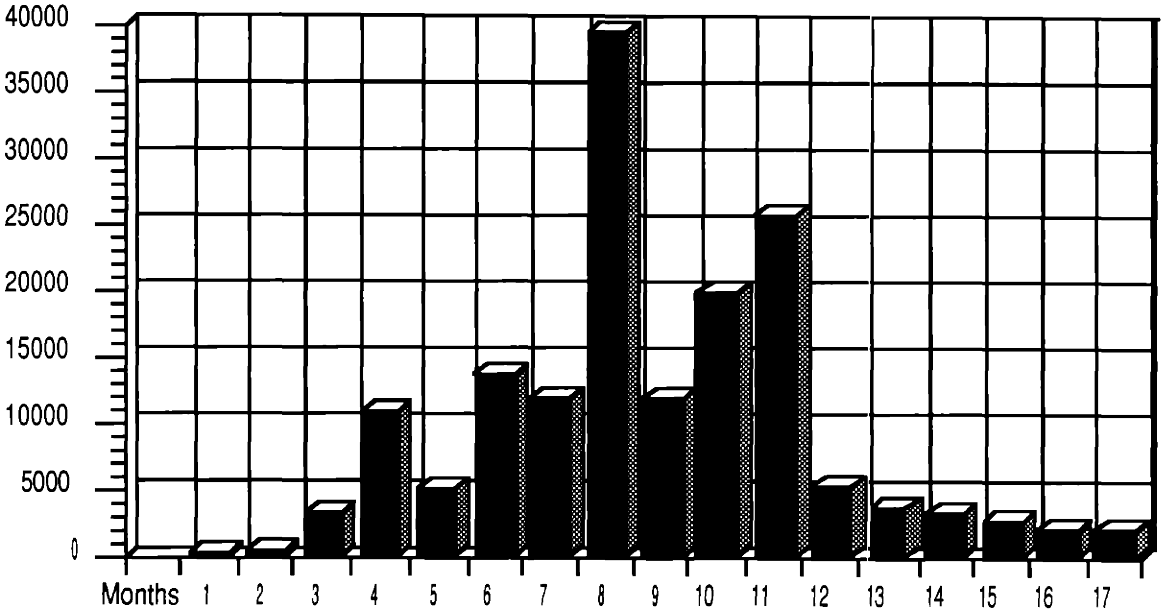
The typical reaction when someone learns of the length of time allocated for component and assembly processing is amazement. Why does it take so long? One possible answer, often postulated, is that because these parts are used on aircraft they require a more complicated fabrication processes. While this is certainly true of some major components (i.e. - wing spars, wing panels), the bulk majority of the parts produced for the aircraft industry are no different than you might find in automotive manufacturing plant. As you might imagine, the actual "touch labor" time[3] that occurs during these flowtimes is very small, less than 5% of the total flowtime. [Copes, 1992] [Shanahan, 1991] Most of the flowtime (90 %), the order is waiting in a queue at some work process.

The Reorder Lead Time (ROLT) specifies the flowtime required to receive delivery of components ordered from external suppliers. For example, if the ROLT for one of Boeing's suppliers is 12 months, Boeing will have to place its orders with that supplier a year before the actual need date. The ROLTs vary from part to part and from supplier to supplier. The current ROLT distribution[4] is shown in Figure 2.5. There are approximately 40,000 part numbers that have eight month ROLTs.

---

[3] Touch labor time refers to the time that the part is actually being transformed by some manufacturing process.

[4] Data provided by Boeing's Materiel Division.



Mean = 9.14 Months

Mode = 8 Months

Fig. 2.5 - Reorder Lead Time (ROLT) for External Suppliers

When ordering action takes place, the order quantity is determined based on the appropriate ordering policy, and the time to release the order is determined by subtracting the required number of days of flowtime from the need date for that component.[5]

Once the order size and date of release have been determined, the order is placed in an electronic queue where it will reside until it is time for its release. The order may be modified while in the queue because of changes in production requirements, bin quantity changes, etc. Modifications to the order may change the order quantity, order release date, or both. After order release, the order priority influences the order processing within the production center. As work orders move through the production center, they are processed in priority order. The primary Boeing order priorities are listed in Table 2.2. Airplane on Ground (AOG) is a priority level that indicates that a customer's airplane will not be operational until the part in question is available. Stores Out of Stock (SOS) indicates that a higher used-on assembly can not be built because of a shortage of one or more detail components. As orders are in work, the order priority may change depending on the load date of the used-on assembly. Within 13 days of assembly load, the order priority changes to Priority 1 from Priority 7.

As an example of how the order control portion of the ordering system functions, assume that an order for a lot of small assemblies consisting of

---

[5] The Order Write Lead Time (OWLT) is also included as part of the planned flowtime. The OWLT varies from a few days to a few weeks; it is used to collect all tools, drawings, etc. required to start actual manufacture of the components.

Table 2.2 - Priority Codes used within Boeing Ordering System

Boeing Priority Codes	**Description**
Priority 6A	AOG
Stores Out of Stock	A part is not available on the scheduled load date of its used-on assembly or installation
Priority 1	Orders with Load Dates that fall from 1-13 days from the current M-day
Priority 7	Orders with Load Dates that fall from 14-55 days from the current M-day

only two details is to be released to a production center and is currently in the order queue. At 55 days prior to order release, the ordering system will check its electronic bin balance to determine if sufficient details exist in the storage bins to complete the upcoming order. If there is not a sufficient quantity for one of the details, the ordering system will check to see if an order is open for the detail in question. If not, an order will be immediately released to the appropriate production center. If an order is open, its priority will be adjusted as the load date for the assembly order approaches.

The Order Location System (OLS) tracks the order as it moves through different production centers. OLS indicates where the order is in the process flow. As parts or assemblies are placed in or removed from storage, the Parts Inventory Control System (PICS) is debited or credited with the correct number of parts. The bin totals are fed back to the SROA system so that it can update its part balance, but the update is not immediate as there is a delay in the update feature between the two systems. If there is a discrepancy between what is in the SROA file and what is in the PICS file, the PICS file information is used because it is assumed to represent what is actually in the bin.

Current Problems within the Ordering System & Their Root Causes[6]

Part shortages present one of the major problems within the production environment. As discussed earlier, a part is considered Stores Out of Stock (SOS) when it is not available on its parent's load date. The major root causes of part shortages are shown in Figure 2.6.

Ordering changes & errors account for the majority of part shortages. There are many different problems that fall into this category. For example, delays in engineering, manufacturing planning, tooling, etc. can lead to late production and detail shortages. Errors in engineering, planning, part card coding, etc. can lead to orders not being placed when they should be.

Product quality is another factor that contributes to part shortages. If parts are rejected and either scrapped or reworked, the timing of this event can lead to shortages if there are not sufficient usable parts in storage. Once the parts are rejected, the ordering system will adjust its balance for that part and place

---

[6] Data for the root cause analysis were collected by a Boeing Preferred Process team established to evaluate the Boeing Ordering System.

additional orders as required, but a shortage condition may remain active until the newly released order is completed.

Process flowtime variation accounts for a portion of the SOS conditions experienced at Boeing. Long queue times, machine downtimes, labor limitations, etc. contribute to variation in the flow of work orders through the production centers. In other cases, committed flows from the engineering

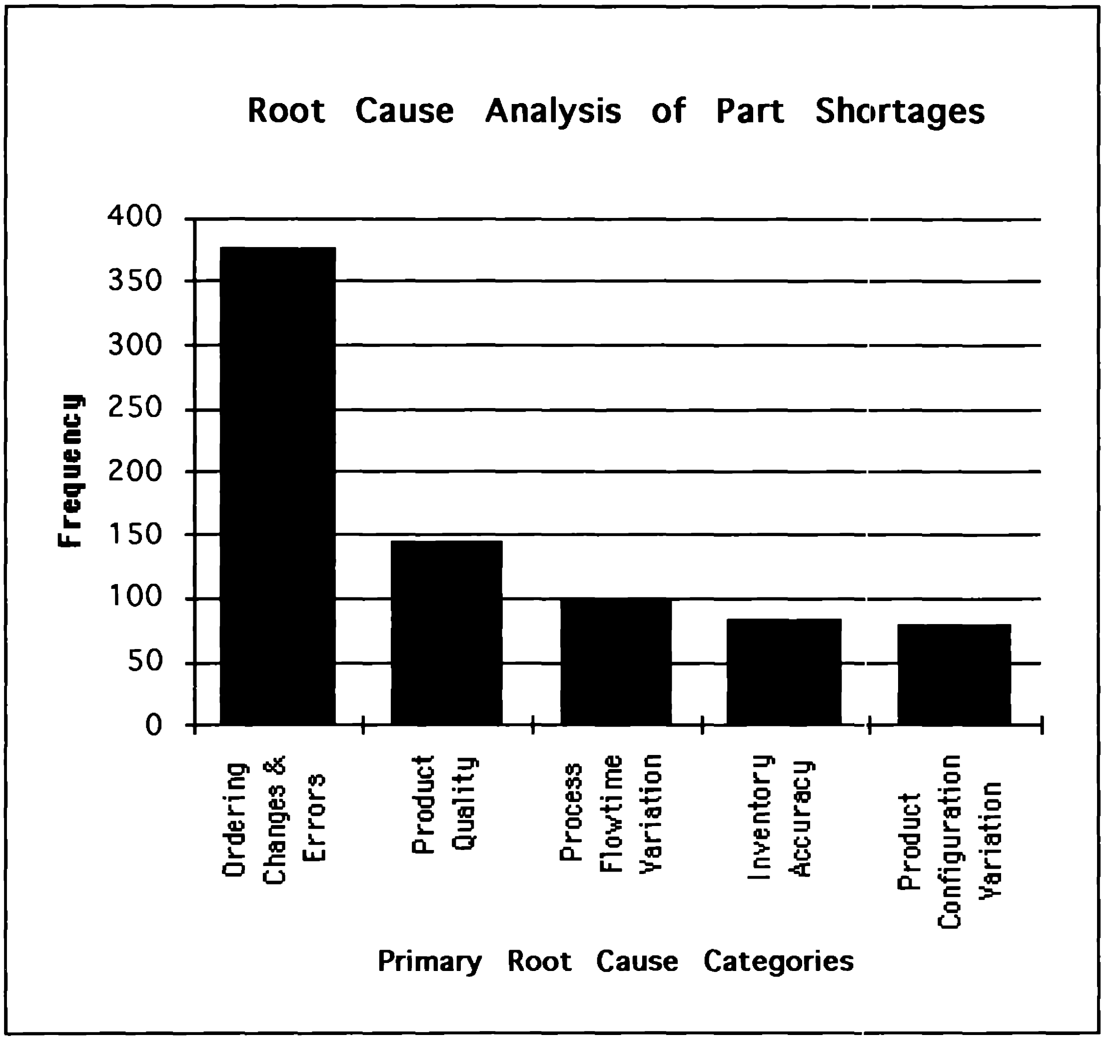


Fig. 2.6 - Root Cause Analysis of Part Shortages

change process are unrealistic, based on current process capabilities, and therefore frequently these committed schedules are not met.

Obviously, inventory accuracy can have a significant impact on the performance of the ordering system. If the bin balance for a detail part is reported to be much higher than it really is, the ordering system will not order additional parts and a shortage condition will result when storekeepers go to pick the details and find that not enough are available.

Product configuration variation leads to frequent changes in the product. When the changes require a quicker response than the system is currently capable of providing, shortages result.

The second major problem experienced in the production environment is excess inventory[7]. After walking through many of the production control areas (PCAs) where Boeing inventory is stored, one sees large stacks of inventory which equates to months, and in many cases years, worth of inventory. Root cause analysis of this problem yielded the results seen in Figure 2.7.

Ordering errors account for a large portion of the excess inventory seen in the production environment. Due to the complexity of the ordering systems and ordering process, human errors occur frequently. For example, when a

---

[7] Excess inventory is defined as inventory that is in excess of requirements or is being received too far in advance of actual need.

manufacturing engineer uses the wrong cutover point[8] for a part procurement, two suppliers of this part will be delivering parts for the same production requirements.

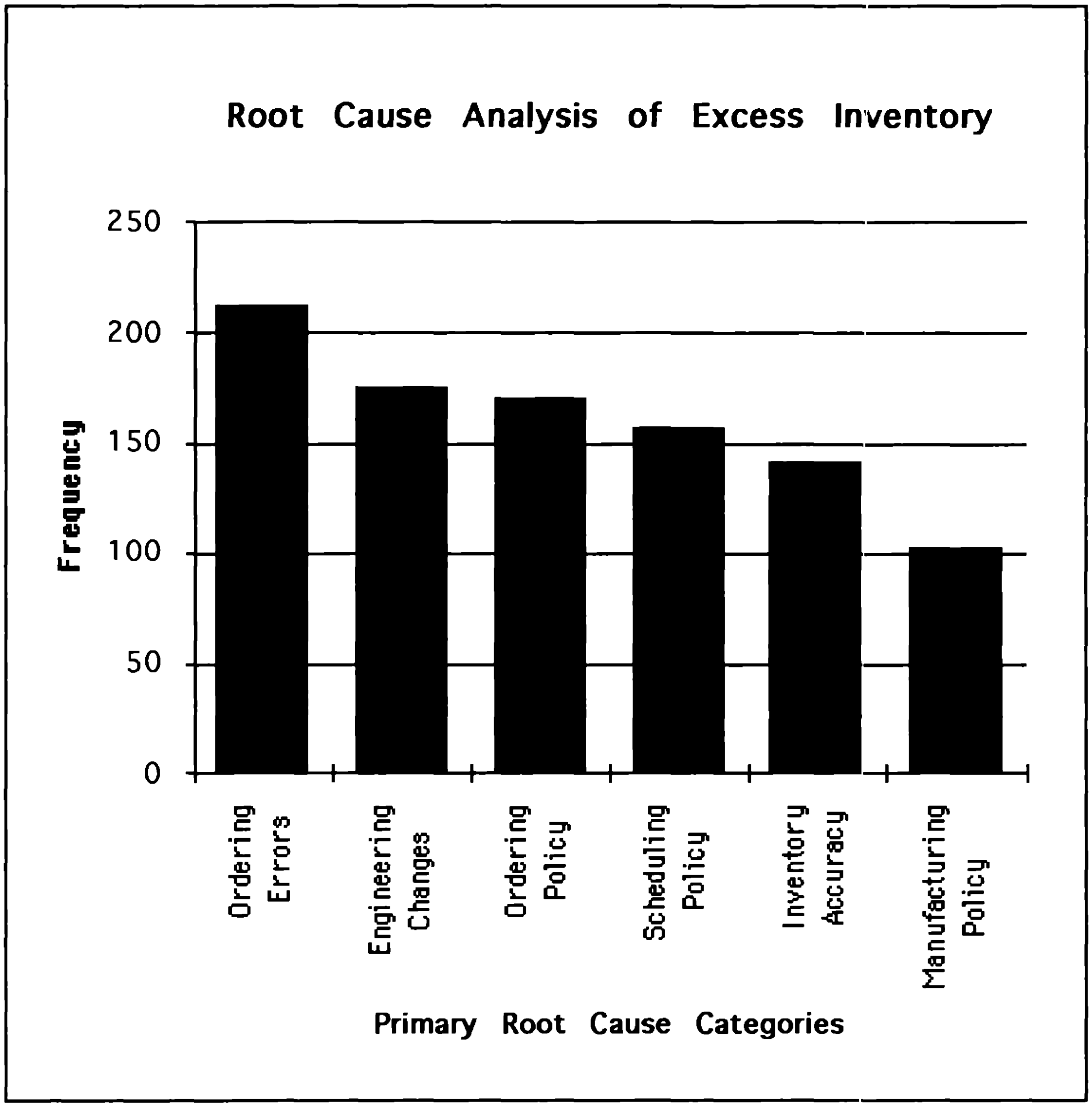


Fig. 2.7 - Root Cause Analysis of Excess Inventory

---

[8] The cutover point is when procurement switches from one source to another.

Engineering changes are also directly responsible for excess inventory generation. The engineering bill of material (EBOM) differs from the manufacturing bill of material (MBOM). When the EBOM calls out one type of procurement for a part, but the MBOM uses another, there will be two sources for the part until the problem is discovered and corrected.

Another common problem is the changing effectivity of engineering drawings. Initially, when a drawing is released, the engineer may tab the drawing for basic effectivity which means that it will apply to all aircraft for a given make and model. At a later time, the engineer determines that the release will not apply to all aircraft and so the drawing becomes tied to a specific configuration. In the meantime, orders have been released and parts have been received to satisfy the original basic effectivity tabbing of the drawing.

Ordering policy within the Scheduled Requirements and Order Analysis (SROA) system generates orders for large lot sizes. Using EOQ and Time Span ordering logic, inventory generation that is 12-13 times the current demand requirements is not uncommon.

The ordering policies between divisions are also not synchronized which creates the problem seen in Figure 2.8. In this example, details supplied by the Sheet Metal Center use an EOQ ordering policy, and the assembly build-up area, which uses these details, uses a Time Span ordering policy. The increment code identifies the part cost with the cost becoming more expensive from B to E. As the figure clearly demonstrates, the supply of the details outpaces the parent assembly build quantity. This lack of ordering

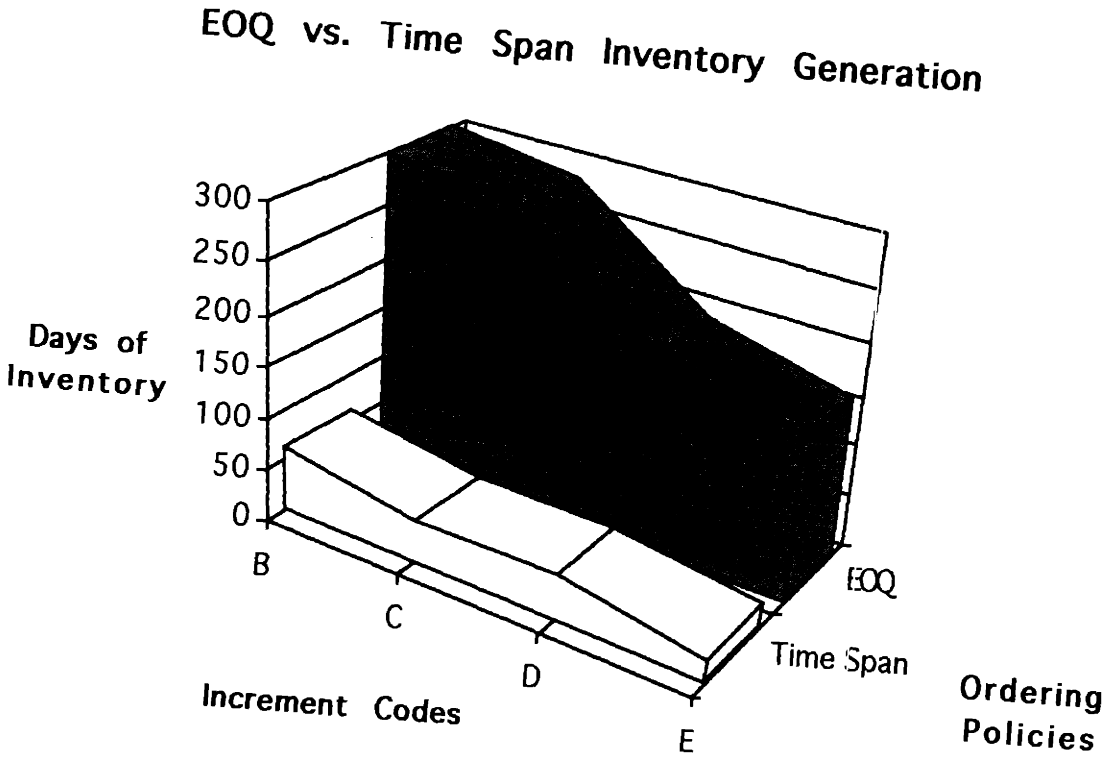
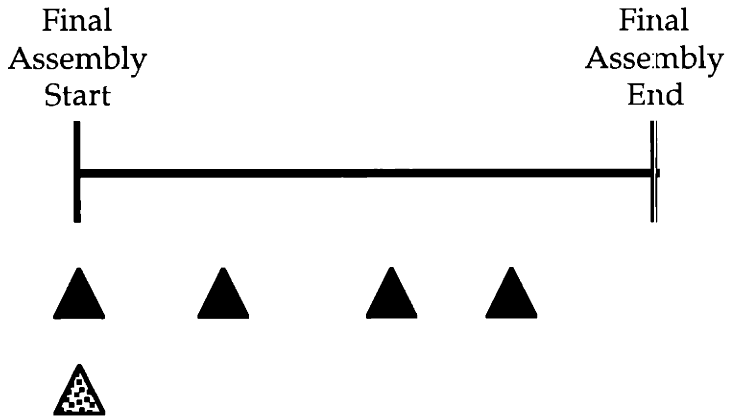


Fig. 2.8 - Ordering Policy and Inventory Generation

policy synchronization leads to inventory that will be held in storage from 150-300 days prior to being depleted.

Within the Boeing ordering system, parts are scheduled to their earliest usage and to the first position within a control code. This scheduling policy leads to inventory being held weeks, and sometimes months, longer than required. For example, if a part is used several places within final assembly, all production requirements for that part will be scheduled to the first usage. The final assembly sequence for an aircraft can be a year in length, and parts that are not needed until the end of final assembly will wait in storage for several months prior to use as shown in Figure 2.9.

Inventory accuracy can cause excess inventory to be generated in much the same way that it can cause part shortages. If there is actually a sufficient quantity of parts in storage but the ordering system receives erroneous data indicating that there is a shortage, excess inventory will be created.



▲ Where Component is Used in Final Assembly.

◬ Where all Final Assembly requirements are delivered.

Fig. 2.9 - Scheduling Parts to First Usage

Finally, the manufacturing policies of a production center can lead to excess inventory generation. A shop may overbuild a component because of perceived economies of scale. In addition, work is sometimes started early because of the desire of management to keep workers busy. [Copes, 1992]

The two major problems, part shortages and excess inventory, and the many elements that cause them to occur provide a static picture of the performance of the Boeing ordering system. A dynamic analysis of the ordering system is required to fully understand how the system operates.

Chapter 3 - System Dynamics Analysis of the Boeing Ordering System

Mathematical modeling has long been used to analyze the dynamic behavior of complex systems. Automatic control has had an impressive history from James Watt's centrifugal governor to the sophisticated flight controls used on jet aircraft. The complexity of these engineering systems required the development of precise mathematical formulations of the system and simulation of system response to various inputs and disturbances. Information gained from these efforts helped in the design of the control elements of those systems.

Social systems, like the Boeing ordering system, are complex nonlinear systems that exhibit dynamic behavior. By developing and using mathematical models of business systems, problems experienced within the system can be better understood and alternative policy actions can be tried to improve overall system performance.

Method Used

System Dynamics modeling describes the general tendencies of the system under study. Is the system stable ? Is it growing, declining, or oscillating ? The primary goal of this method is to uncover the time varying pattern of the system without attempting to provide a precise forecast of future system behavior.

System dynamics analysis draws on several different fields for model construction - control engineering, cybernetics, and organizational theory. The basic assumption of this method is that the complex behavior often seen within systems is a result of their causal structure. Feedback, the transmission and return of information, plays a central role in this analysis. There are two types of feedback- positive and negative. Positive, or self-reinforcing, loops like the one described in Figure 3.1 lead to exponential growth within the system. In this example, capital is invested and earns interest and is then reinvested to earn more interest. The graph shows the exponential increase common for positive feedback systems.

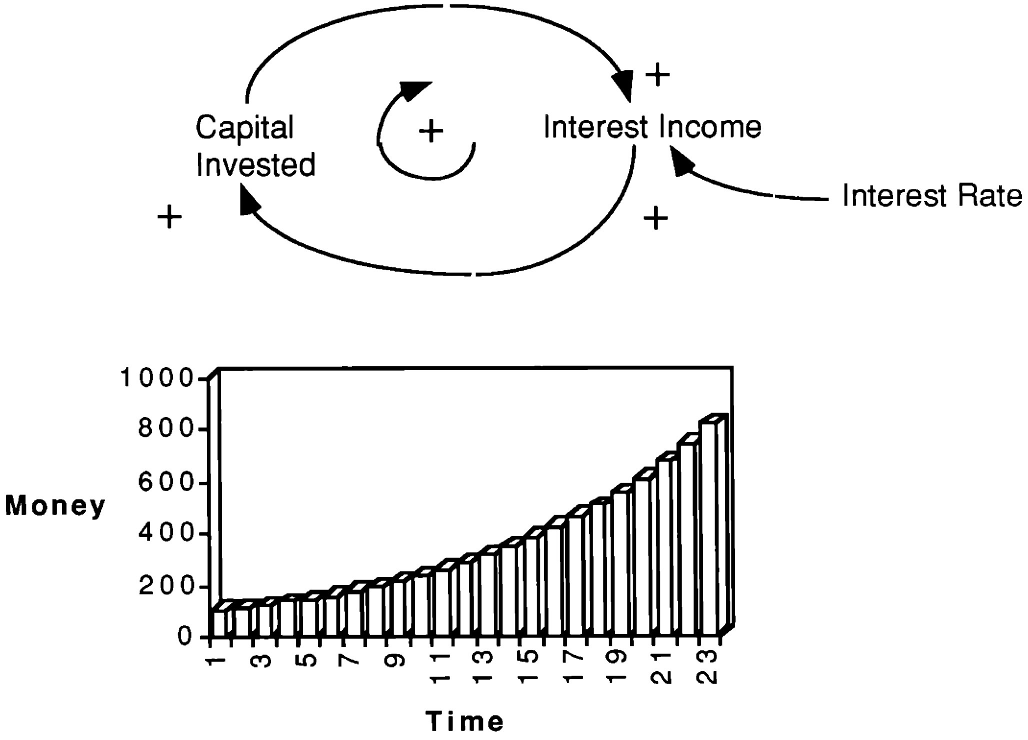


Figure 3.1- Positive Feedback Loops

The counterpart to positive feedback loops are negative, or goal-seeking, loops like the one shown in Figure 3.2. In this classic example, a thermostat attempts to maintain a desired temperature by turning on a heat source which then changes the environment in the room. There is a delay between the time that the heat source is activated and the room temperature starts to change. When the desired temperature is reached, the heat source is deactivated. The characteristic behavior of a system governed by a negative feedback loop is shown in the Fig. 3.2. The actual value tracks a target level, and the amount of error is largely dependent on the delays in the system.

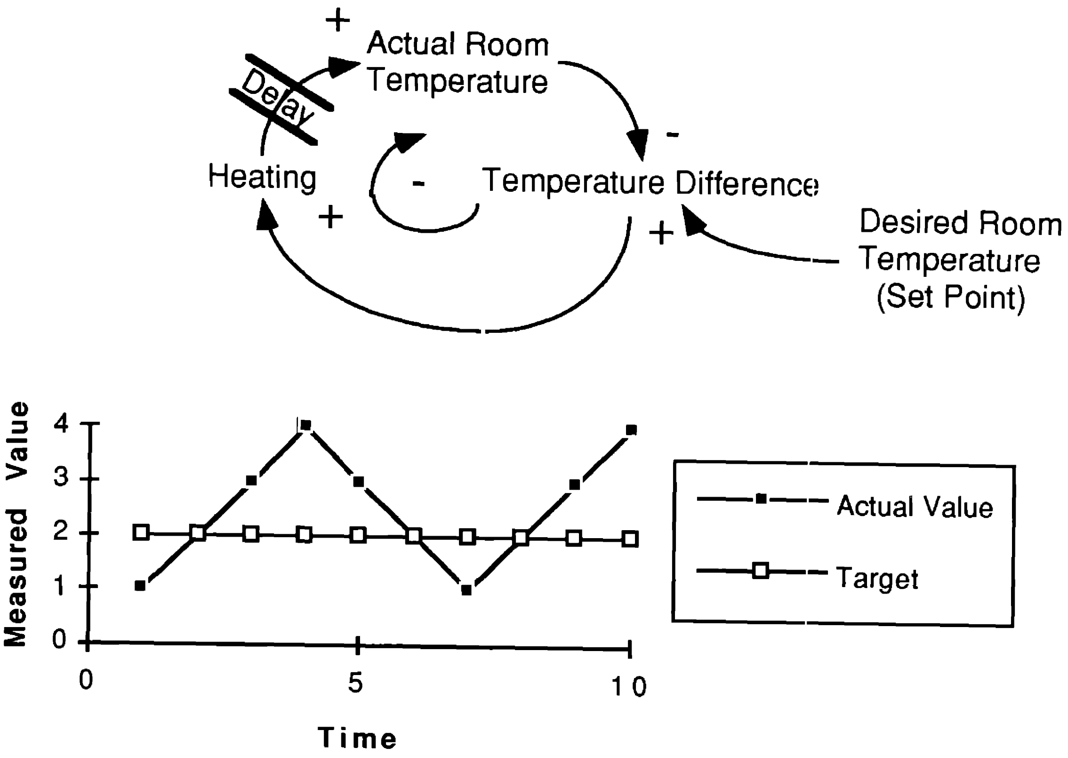
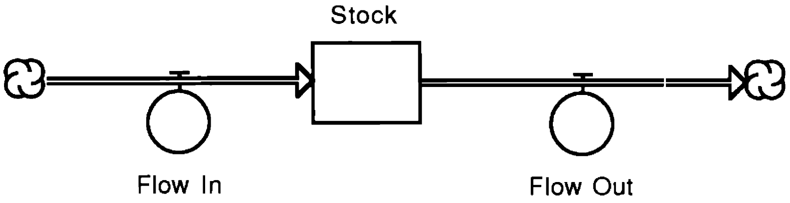


Figure 3.2 - Negative Feedback Loops

System dynamics models are a combination of positive and negative feedback loops. All information-feedback systems consist of three elements: 1) structure, 2) delays, and 3) amplification. The structure shows how all the parts of the system are linked together. Delays exist in collecting information, deciphering it, making decisions, and deciding to act on those decisions. Amplification occurs in systems when the output response is stronger than the input that caused it.

For the purpose of this research, the mathematical model used to represent the ordering system consists of a collection of difference equations using a first-order Euler integration method.[1] A higher order integration scheme such as a second or fourth order Runge-Kutta integrator was not required because of the discrete nature of the system being studied. Examples of the structural elements[2] that are used to build system dynamics models are shown in Figure 3.3 along with the corresponding difference equation that describes it. Stocks are the accumulations within the system. As an example, inventory is a stock whose current value is the accumulated difference between the inflows and outflows. Flows represent the current instantaneous rate (over the dt) between the stocks. Flows represent the dynamic activity that occurs in the system, and stocks represent the resulting state of that activity.



Stock(t) = Stock(t-dt) + (Flow In - Flow Out)*dt

Figure 3.3 - Model Structural Elements

Approach Used for Analysis

The Boeing Ordering System controls hundreds of thousands of parts and consists of many different hardware platforms and software applications to perform this function. The model developed in this thesis in no way attempts

---

[1] A software package called Ithink™ developed by High Performance Systems, Inc. was used to build the model.

[2] For a further review of model elements, please refer to Appendix A-1.

to capture all elements of the complete system. The area of concentration was the Scheduled Requirements and Order Analysis (SROA) portion of the system that is responsible for ordering a variety of sheet metal and machined parts from internal and external suppliers. Furthermore, the model recreates only a cross section of this area.

To facilitate the development of the model, a small assembly was chosen for analysis. This assembly is a structural member used on the Boeing 757 #2 passenger door frame. It consists of four detail components: stringer, web, doubler, and clip.[3] The stringer and doubler[4] are produced in the Sheet Metal Center which is part of Boeing's Fabrication Division. The other two components are manufactured by external suppliers. All four of the components are delivered to the Renton Lot Time area where they are placed in storage. From storage, the details are kitted and built into subassemblies which are then delivered to a storage area in final assembly. In final assembly, the subassemblies are installed as part of the 757 #2 passenger door frame.

While this is only one subassembly, the model structure is applicable to many different details and subassemblies built by Boeing. For example, the model structure of subassembly ordering action is applicable to many types of subassemblies built by the Renton and Everett Lot Time areas.

---

[3] For a further review of data on assembly components, please refer to Appendix B-2.

[4] The doubler has recently been off-loaded to an outside supplier. All data on the doubler is based on internal production. In the model, it is treated as an internally produced component.

Model Overview

Figure 3.4 provides an overview of the model developed for this project. As shown in the figure, five basic sectors are included in the model: 1) Major Assembly Operations, 2) Ordering Action, 3) Supplier Fabrication, 4) Component Assembly, and 5) Detail Fabrication. Portions of the model structure will be discussed within the body of the thesis, but the complete model (i.e. - diagrams and equations) can be seen in Appendix A-2 & A-3.

1. Major Assembly Operations

Major Assembly Operations drive the demand rate for the subassembly. The subassemblies are stored in Production Control Area 46 in final assembly. The current aircraft production rate determines how fast the inventory level for this subassembly is depleted. The aircraft production rate from final assembly will also drive ordering action for the details used in the subassembly.

2. Ordering Action

This sector consists of the following activities: 1) order generation, 2) order accounting, and 3) order prioritization and expediting. The primary task of the ordering sector is to generate orders for the fabrication areas (internal and external) and the Renton Lot Time assembly area.

As an example of model structure, elements of the ordering sector are shown in Figures 3.5a-c. In Fig. 3.5a, the order release process for Renton Lot Time subassemblies is diagrammed. An order is released to production based on the order release signal. This signal can be the result of an automated or manual process. SROA will automatically release an order with sufficient schedule

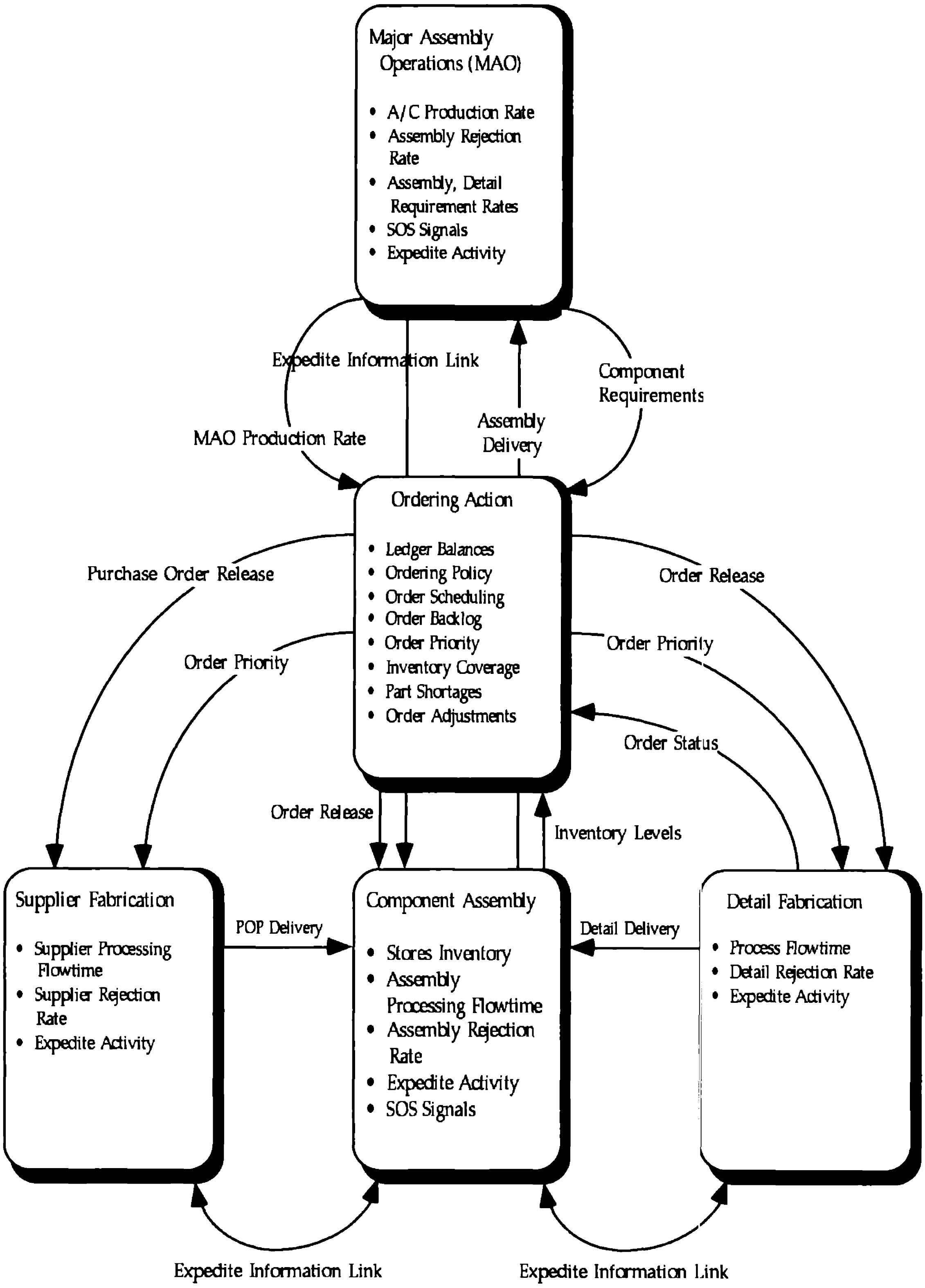


Figure 3.4 - Model Overview

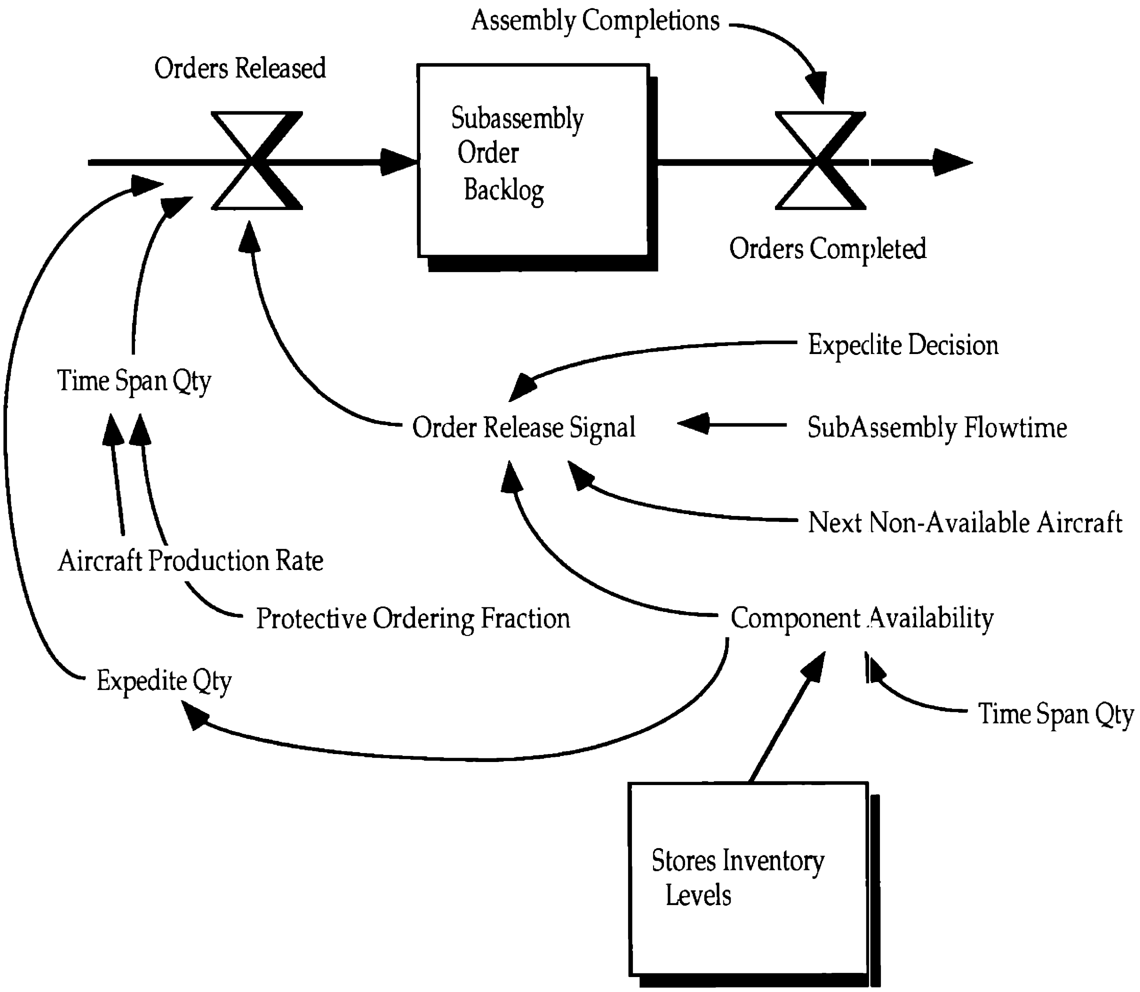


Figure 3.5a - Ordering Sector - Renton Lot Time

advance to supply subassemblies in support of the next non-available aircraft (i.e. - the next aircraft after all current inventory has been exhausted). This decision is based on the final assembly load date for the next non-available aircraft and the current subassembly flowtime allocation. For normal order release within the Renton Lot Time area, the order quantity is determined by the current time span parameter being used, the current aircraft production rate, and any protective ordering that is being used. Once the order is released, the subassembly order backlog increases by the order amount, and this backlog will not be cleared until the subassemblies are completed.

Frequently there are not sufficient components within current inventory to build the entire lot as dictated by the ordering policy. When this occurs, the expediter within the Lot Time area has to determine if a portion of the order should be built or if the order should wait for incoming orders to replenish the delinquent detail's inventory level. If a decision is made not to wait for the incoming order, the expediter will revise the order and release it to the floor with the new expedited quantity penciled in.

Elements of the expedite decision can be seen in Figure 3.5b. The expediter determines how many components are available, and the type and quantity of any details that are SOS. Based on when the SOS details will be delivered and the timing of the next non-available aircraft, the expediter decides whether or not to build the order with some fraction of the available components.

Information on when the next non-available aircraft will occur is contained within the SROA ledger logic of the ordering system. The model representation of this structure can be seen in Fig 3.5c . The ledger balance contains the number of subassemblies that are currently available to satisfy production. New deliveries add to the balance, and the production rate in Control Code 352 decrements the number of subassemblies that are available. Adjustments to the ledger balance occur as updates from the Parts Inventory Control System (PICS) are transmitted. These adjustments will add or subtract from the current ledger balance. Based on the current ledger balance and the known production rate, the next non-available aircraft can be determined.

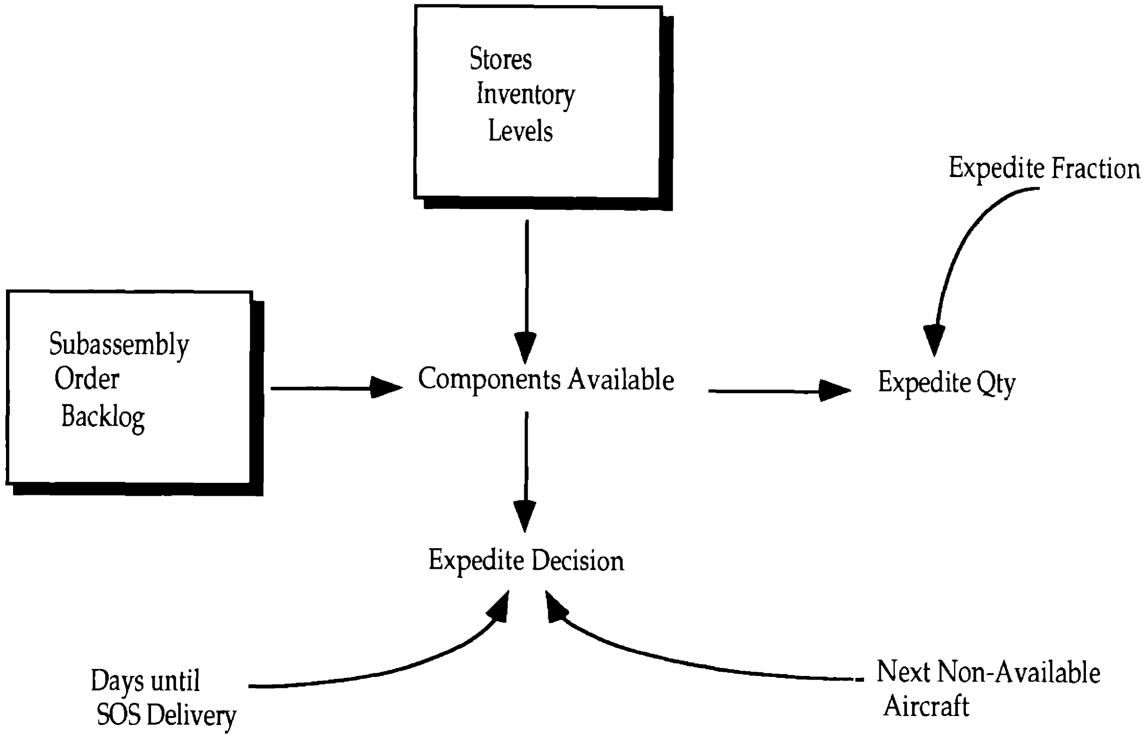


Figure 3.5b - Lot Time Expedite Actions

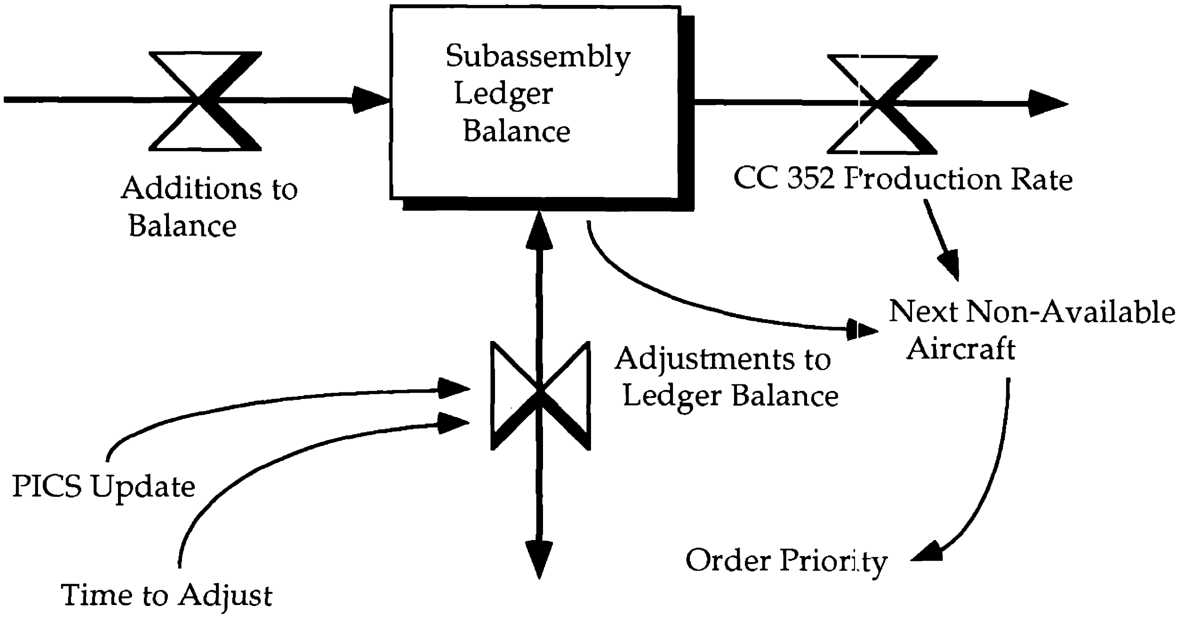


Figure 3.5c - Subassembly Ledger Balance

The order priority for subassembly orders is determined by the proximity of the current manufacturing day to the next non-availability date. As the non-availability date approaches, the order priority increases.

3. Supplier Fabrication

Some of the orders that are generated go to suppliers. SROA passes the order requirements to Material Division which will then place the order with the supplier. The Reorder Lead Time (ROLT) controls the order release date. Imbedded within the ROLT is the administrative time to place the order, the fabrication time, and the transportation time. Product quality from the suppliers determines the number of parts per shipment that are acceptable. Late orders will be expedited by the appropriate Buyer in the Materiel Division.

4. Detail Fabrication

The ordering system will release orders to the fabrication area and will change the order priority as appropriate. If the order is SOS, which means that it is preventing the build of a parent assembly, it may be expedited. There are two components of the flowtime in the fabrication area: queue time and processing time. For the purposes of this model, the processing time is aggregated and represented by a normal probability distribution.

5. Component Assembly

The Renton Lot Time area is where the detail components are stored and later assembled. The model representation of this sector can be seen in Fig. 3.6. Detail components arrive from internal and external suppliers and are stored in

PCA 370 . Adjustments to the stores level are made for a variety of reasons. Some parts are rejected for quality problems or engineering changes

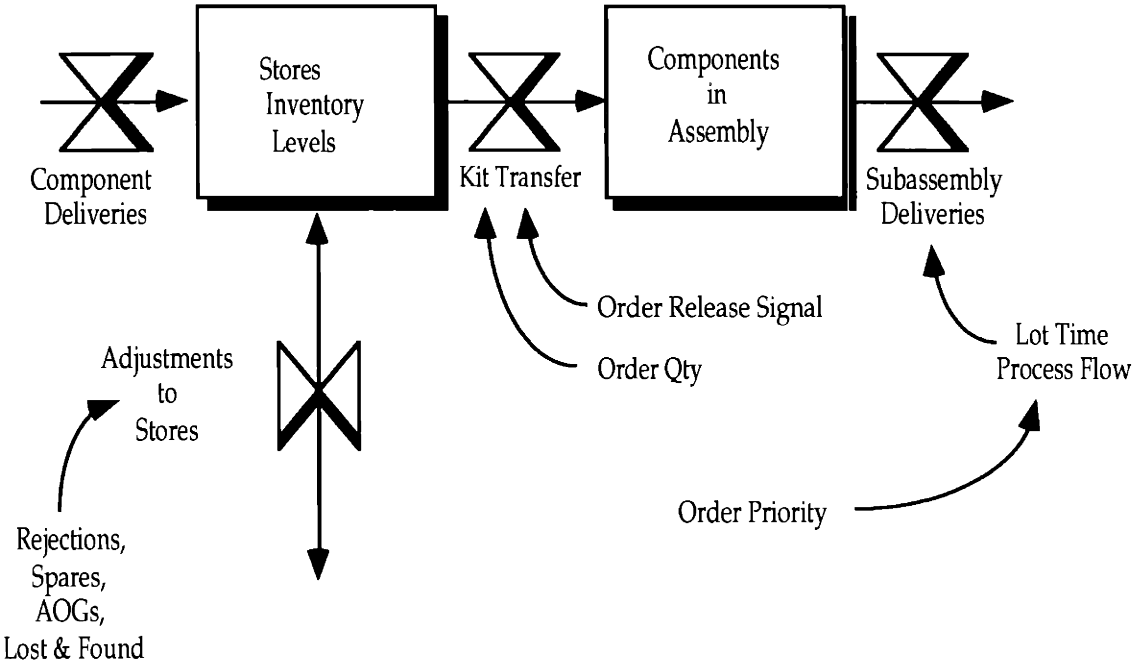


Figure 3.6 - Component Assembly Operations

(components pulled for rework also fall into this category). Occasionally spares and AOG activity will pull components from the storage area to support aircraft in the field. In addition, inventory is sometimes misplaced leading to adjustments.

Kits are filled with detail components when order release signals are generated. These signals can be either manually or automatically generated as discussed previously. The number of details used to fill the kits are based on the quantity specified on the order. Once the order is kitted it is taken out into the assembly area where it resides until it is worked on.

The model aggregates the processing time combining both the queue and actual touch labor time. The process flow time is influenced by the order priority. As the order priority increases, the processing time decreases. After processing is complete, the subassemblies are delivered to PCA 46 where they are stored.

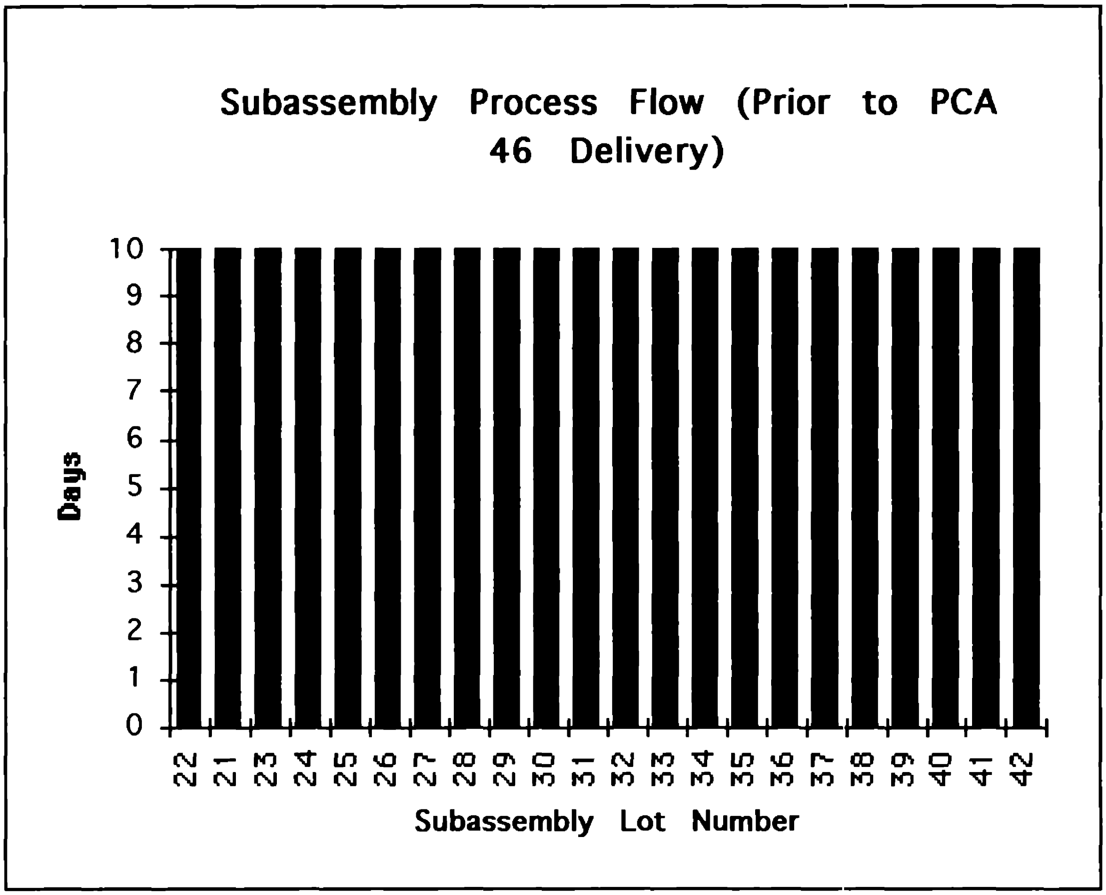
Data Sources Utilized

The model relied on data from several different sources. Model structure was the result of direct observation and interviews with employees within each of the areas represented in the model. Historical data of ordering actions for each of the details and the assembly were collected, where available, to determine the behavior of the system. Appendix B-3 contains a sample of the type of internal data used to determine historical performance and define model parameters.

A persistent problem throughout this project was the quality and insight provided by the available data. Data is abundant within Boeing computer systems and archives, but in many cases the data do more to confuse than to clarify how a given system is performing. Figure 3.7 contains an example of the problem of data quality within Boeing databases. In this example, data meant to convey the actual processing flow time within the Renton Lot Time area is clearly bogus.

The chart shown in Figure 3.7 was created from data taken out of the Boeing Ordering System. The subassembly flowtime shown in this chart is based on the following calculation.

Actual Completion - Actual Start = Subassembly Flow Days



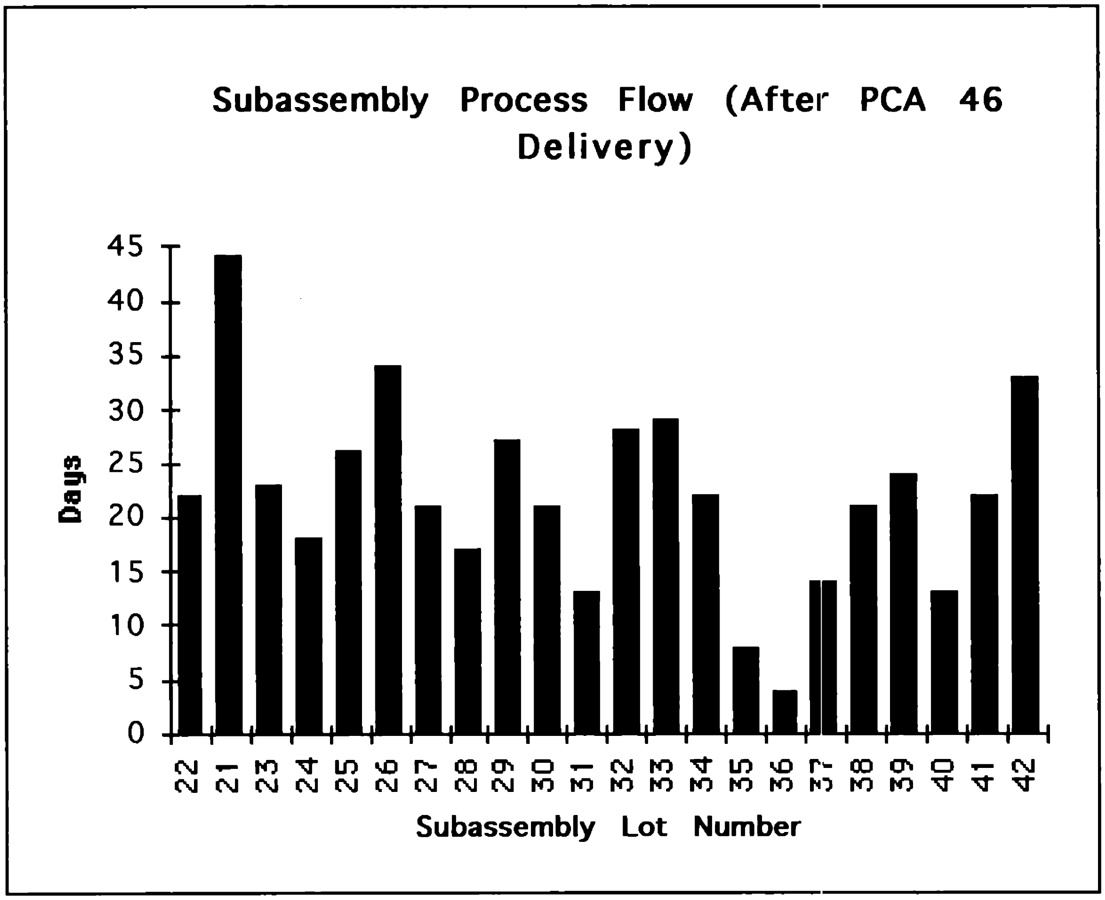
Mean = 10 Standard Deviation = 0

Figure 3.7 - Subassembly Process Flow (Prior to PCA 46 Delivery)

The process flow time for Lots 21-42 above were all ten days. Obviously a constant (in this case 10 days) is added to the actual start date. The Puget Sound Flow standard for all Lot Time processing is ten days, and therefore it appears that the Puget Sound Flow standard is just being added to the actual start date of each job that passes through the Renton Lot Time area. This calculation was not used to determine the subassembly process flow times for this study.

A more accurate representation of the subassembly flow days, shown in Figure 3.8, was based on the following calculation.

Actual Delivery to PCA 46-Actual Start in Lot Time = Subassembly Flow Days



Mean = 22 Standard Deviation = 8.92

Figure 3.8 - Subassembly Process Flow (after PCA 46 delivery)

Based on discussions with personnel responsible for subassembly processing, the processing flow times shown above are more representative of what actually occurs in Renton Lot Time Area. The mean and standard deviation of this sample were used as a starting point for model parameter values that represent the processing flow distribution in the Renton Lot Time area for assembly operations.

It is difficult to see how general productivity will improve when simple and accurate measures of current performance are not readily available. The problem of data overload and information scarcity plagues many companies, and Boeing is no different. The ease of generating data with today's computer systems causes Information Technology planners to err on the side of abundance. What is needed is a synthesis of appropriate data into simple metrics which clearly indicate system performance.

None the less, there was sufficient data of high enough quality to build a system dynamics model representing the Boeing Ordering System. With subsequent revisions of the model by Boeing employees or other interested parties, new and more accurate data can be used to improve upon the model. In Table 3.1, the major parametric and variable assumptions used in the model can be seen.

The parameter values were determined in a variety of ways. In some cases, sufficient information was available to determine the values directly. As an example, the Boeing production rates are well known. Many of the values were determined in this manner. Other parameters are based on Boeing policy guidance. Puget Sound Flowtimes fall into this category. Other parameters were estimated. The expedite timing threshold is an example of this type of parameter. When confronted with an impending shortage to final assembly, how long will an expediter wait for a delinquent detail delivery before he decides to build less than the quantity of the original order ?

Table 3.1 - Major Model Parameters & Variable Values Used

Description	Value Used
Aircraft Production Rate (aircraft produced every x days)	2.5 - 4
Rejection & Rework Rate (percent of order scrapped)	0-30%
Stringer Economic Order Quantity Values (units per order)	35-200
Average Stringer Processing Time (days)	30
Puget Sound Flowtime - Stringer (days)	71
Doubler Time Span (days)	252
Average Doubler Processing Time (days)	45
Puget Sound Flowtime - Doubler (days)	83
Component Rate of Use Values (unitless)	1 - 1.5
Priority 1 Time Fraction (percent of normal flow)	.9
SOS Time Fraction (percent of normal flow)	.8
Expedite Quantity Fraction (percent of original order)	.25
Expedite Flowtime (days)	5
Average Subassembly Processing Time (days)	22
Subassembly Time Span (days)	42
Puget Sound Flowtime - Lot Time Subassembly (days)	30
Expedite Timing Threshold (days until SOS)	10
Supplier Reorder Lead Times (days)	180
Supplier Order Quantity (days)	252
Purchased Outside Production Schedule Advance (days)	42

A glossary providing a brief definition of the parameters can be seen in Table B-3.3 of Appendix B-3. For a complete listing of the source of information for determining parameter values, please refer to Table B-3.4 in Appendix B-3. With the major structural representation defined and parameter values identified, the model had to be validated. To validate the model several tests were performed. The specific tests used can be seen in Appendix A-4.

No single test validates a system dynamics model; in fact the model is never completely validated. [Forrester & Senge (1981), Richardson (1981), Sterman

(1983)] I encourage the reader, especially Boeing employees, to review the model structure and assumptions with a very critical eye. Undoubtedly as the model evolves, an enhanced representation of system structure and more accurate parameter values will be developed. It is my hope that the model will be a living entity that enhances the user's understanding of how the Boeing Ordering System functions. My goal, all along, has been to produce something that will in a word be - useful.

Chapter 4 - Results of Analysis

Base Run

A base run simulation using the model developed for this project was run to determine the model's accuracy in replicating the reference behavior of the Boeing Ordering System. The simulation time span covers approximately four years of Boeing 757 production. A comparison of the simulation coverage period with the actual Boeing manufacturing calendar can be seen in Table 4.1.

Table 4.1 - Simulation Time Correspondence to Boeing M-days

Simulation Time	Calendar Day	Boeing Manufacturing Day
Start = 0	December 16, 1988	9536
Complete = 980	November 20, 1992	10,516

There are several points of comparison between the model and historical performance of the Boeing Ordering System. The first concerns how accurately the model recreates the quantity of product produced. Secondly, the model should generate deliveries of components with roughly the same sequencing as the actual system. Point prediction is not required to satisfy this requirement, but the model should, however, generate orders leading to deliveries with a similar periodicity. Finally, the model should generate problem behavior seen in the existing system, such as inventory stockouts.

A comparison between the cumulative product and stockouts produced within the actual system and the model can be seen in Table 4.2. The absolute percent of error between model results and historical data is listed in Table 4.3. For a more detailed depiction of model performance, please review Fig. 4.1-4.15 at the end of this chapter. The periodicity of component replenishment can be seen in Fig. 4.1a-4.5a. The cumulative product produced can be seen in Fig. 4-1b-4.5b. As identified in Table 4.2, the number of stockouts generated matches that indicated from actual data. Stockout timing for subassemblies and stringers can be seen in Fig. 4.6 and Fig. 4.7. Inventory stock levels, for all components, can be seen in Fig. 4-8 through Fig. 4-12.

In addition to quantity levels and delivery periodicity, information on the average cycle times of subassembly processing can be seen in Fig. 4.13. In Fig. 4.14, the percentage of subassembly cycle time that the order is in a Priority 1 or SOS condition is traced. These comparisons provide a general comparison of model results with historical data. A more quantitative analysis of model performance is presented next.

To establish the quantitative accuracy of the model, statistical hypothesis testing was used to evaluate model results. An unpaired t-Test was used to determine if the model and historical mean values from the data were the same. [Montgomery, 1991] The statistical hypothesis used to evaluate the mean quantity level of component deliveries can be seen below.

H_0: Model component μ quantity = Historical component μ quantity

H_1: Model component μ quantity <> Historical component μ quantity

Table 4.2 - Comparison of Historical Data vs. Model Output

Comparison	Historical Data	Model Generated Data
Cumulative Subassemblies	296	303
Cumulative Stringers	381	375
Cumulative Doublers	379	368
Cumulative Clips	345	351
Cumulative Webs	344	347
Subassembly Stockouts	3	3
Stringer Stockouts	2	2

Table 4.3 - Absolute Percent Error of Model Output from Historical Data

Comparison	Absolute Percent Error
Cumulative Subassemblies	2 %
Cumulative Stringers	2 %
Cumulative Doublers	3 %
Cumulative Clips	2 %
Cumulative Webs	1 %
Subassembly Stockouts	0 %
Stringer Stockouts	0 %

H_0, the null hypothesis, is rejected if: $|t| > t_{\alpha/2,\, n1+n2-2}$. The alpha level, α, represents the probability of Type I error, rejecting the null hypothesis when the null hypothesis is true.[1] For this testing the alpha level was chosen to be .01, which means that there is a 1% chance of falsely rejecting the null hypothesis. Using standard tables for the t-distribution, the most stringent criteria level for the sample sizes and alpha value chosen is shown below.

$$t_{.01/2,\, 22+22-2} = \mathbf{2.7}$$

To reject the null hypothesis, the t value must be larger than 2.7. In Table 4.4, the t values for component quantity levels are determined. The t-values, for all cases, are very close to zero, and therefore we fail to reject the null hypothesis that the means are equal.

To examine the accuracy of the delivery periodicity, the following statistical hypothesis was formulated.

H_0: Model μ delivery period = Historical μ delivery period
H_1: Model μ delivery period <> Historical μ delivery period

In Table 4.5, the t values for the mean delivery periodicity are calculated. The t-values, for all cases, are very close to zero. Based on the fact that all t-values are less than 2.7, we once again fail to reject the null hypothesis that the means are equal.

---

[1] Conversely, the Beta Level, β represents Type II error, failing to reject the null hypothesis when it is false. The power of the test $(1 - \beta)$ is directly related to the number of samples used in the analysis. The power increases with increasing sample size.

Table 4.4 - t-Test on Component Quantity Levels

t-Test on Quantity	Webs	Clips	Doublers	Stringers	SAs
n1 - Sample Size (Actuals)	3.00	4.00	3.00	5.00	22.00
n2 - Sample Size (Model)	3.00	4.00	3.00	5.00	22.00
S1 - Standard Deviation (Actuals)	11.55	12.69	28.50	7.73	6.72
S2 - Standard Deviation (Model)	15.63	15.11	12.12	5.68	5.82
Sp - Pooled Variance	188.82	194.61	479.61	45.98	39.50
X1 - Mean (Actuals)	98.33	83.25	109.67	72.80	13.23
X2 - Mean (Model)	99.33	84.75	106.00	71.60	12.91
t value	-0.01	-0.01	0.01	0.04	0.03

Table 4.5 - t-Test on Component Delivery Periodicity

t-Test on Periodicity	Webs	Clips	Doublers	Stringers	SAs
n1 - Sample Size (Actuals)	3.00	4.00	3.00	5.00	22.00
n2 - Sample Size (Model)	3.00	4.00	3.00	5.00	22.00
S1 - Standard Deviation (Actuals)	25.46	5.69	57.28	32.88	16.46
S2 - Standard Deviation (Model)	5.66	20.60	36.06	18.76	22.62
Sp - Pooled Variance	340.02	228.37	2290.66	716.52	391.30
X1 - Mean (Actuals)	299.00	273.67	296.50	228.25	42.82
X2 - Mean (Model)	314.00	251.33	285.50	234.00	42.59
t value	-0.05	0.14	0.01	-0.01	0.00

An additional statistical test was performed on the model results. The Mean Absolute Percent Error (MAPE) determines the average deviation of a parameter from its actual value over the number of samples taken. The exact formulation can be seen below.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n} |S_t - A_t|$$

where

S_t - Simulated model value

A_t - Actual value

The results of this analysis can be seen in Table 4.6. The MAPE is a relative measure of performance. Based on a comparison of all component quantity levels and delivery periodicity, we see that doubler performance is weaker than that of the other components. The model results for subassembly quantity and delivery delta, on the other hand, track very close to the actual historical performance. A rank ordering or component performance can be seen in Table 4.7 for both tests.

Table 4.6 - MAPE on Component Quantity Levels & Delivery Deltas

MAPE	Webs	Clips	Doublers	Stringers	SAs
Quantity	6.33	7.00	15.67	4.00	1.32
Delivery Delta	11.00	23.25	46.33	26.00	13.14

Table 4.7 - Rank Ordering of MAPE Performance
(from Best to Worst)

Quantity	Delivery Delta
Subassembly	Webs
Stringers	Subassembly
Webs	Clips
Clips	Stringers
Doublers	Doublers

Conclusion

Based on the graphical results, percent error, and t-tests evaluation, the model accurately represents the reference behavior of the system under evaluation. The MAPE results indicate that modifications of doubler parameter values might improve model response for this component. As was stated in the previous chapter, the model is never completely validated. Even though the statistical analysis indicates a very close fit between model results and historical data, the accuracy of the model will continue to improve as more knowledge is gained about actual system operation.

While there is room for improvement in how the model recreates the behavior, the model should prove to be a good preliminary tool for policy analysis. Policy analysis will allow for further validation of the model. If policies are attempted which lead to a clearly erroneous response, the model structure will be called into question, and refinements will be required.

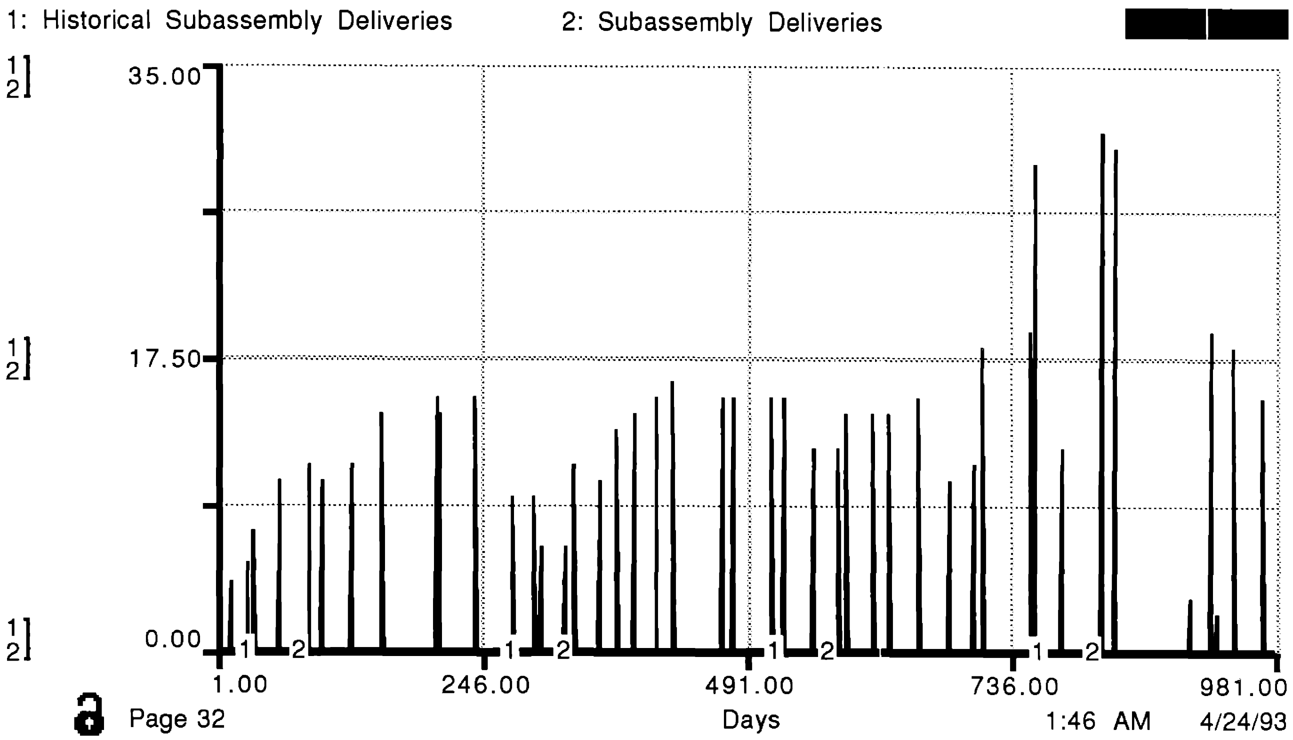


Figure 4.1a - Historical vs. Simulated Subassembly Deliveries

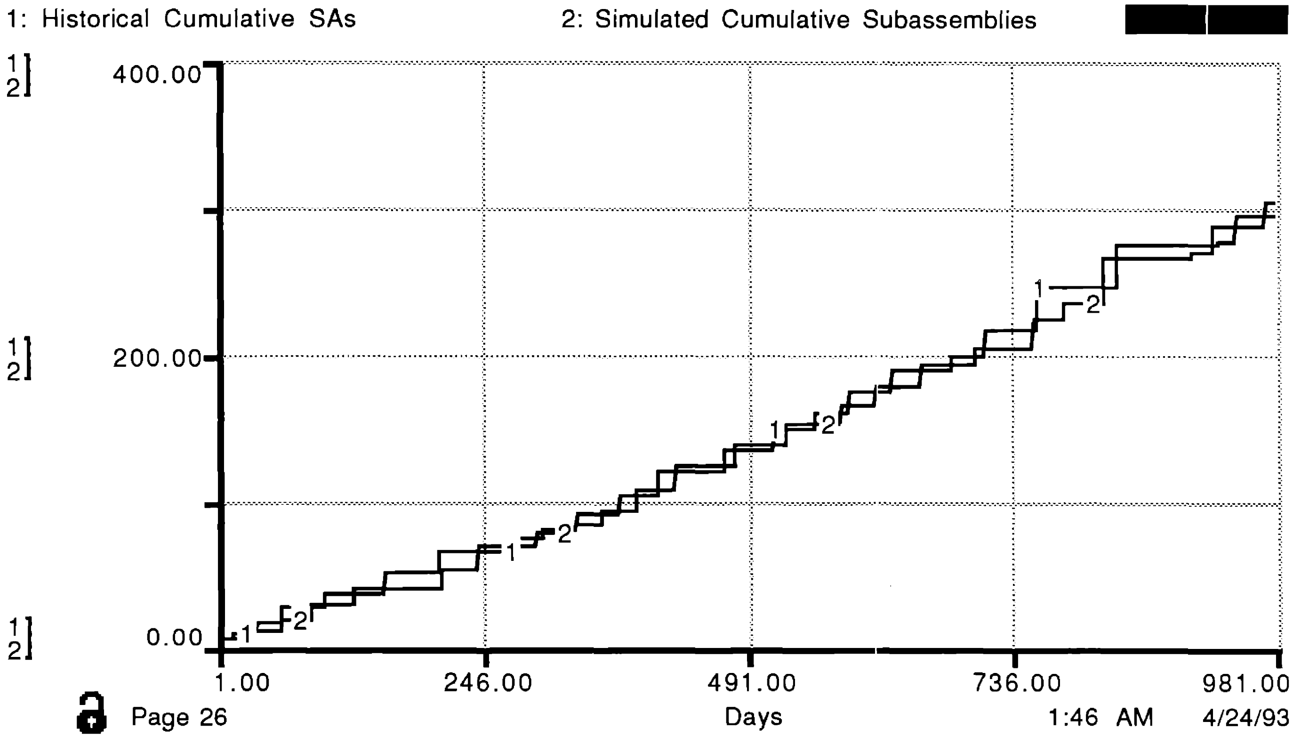


Figure 4.1b - Historical vs. Simulated Cumulative Subassembly Deliveries

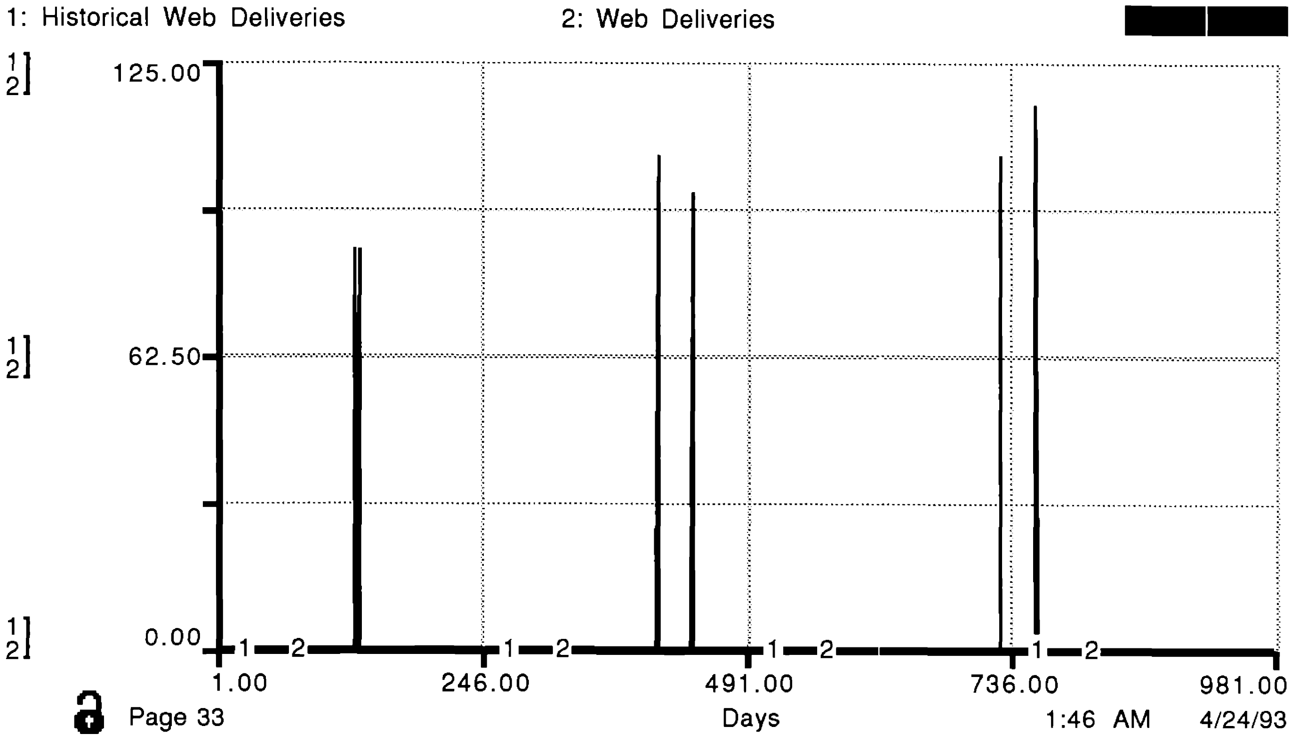


Figure 4.2a - Historical vs. Simulated Web Deliveries

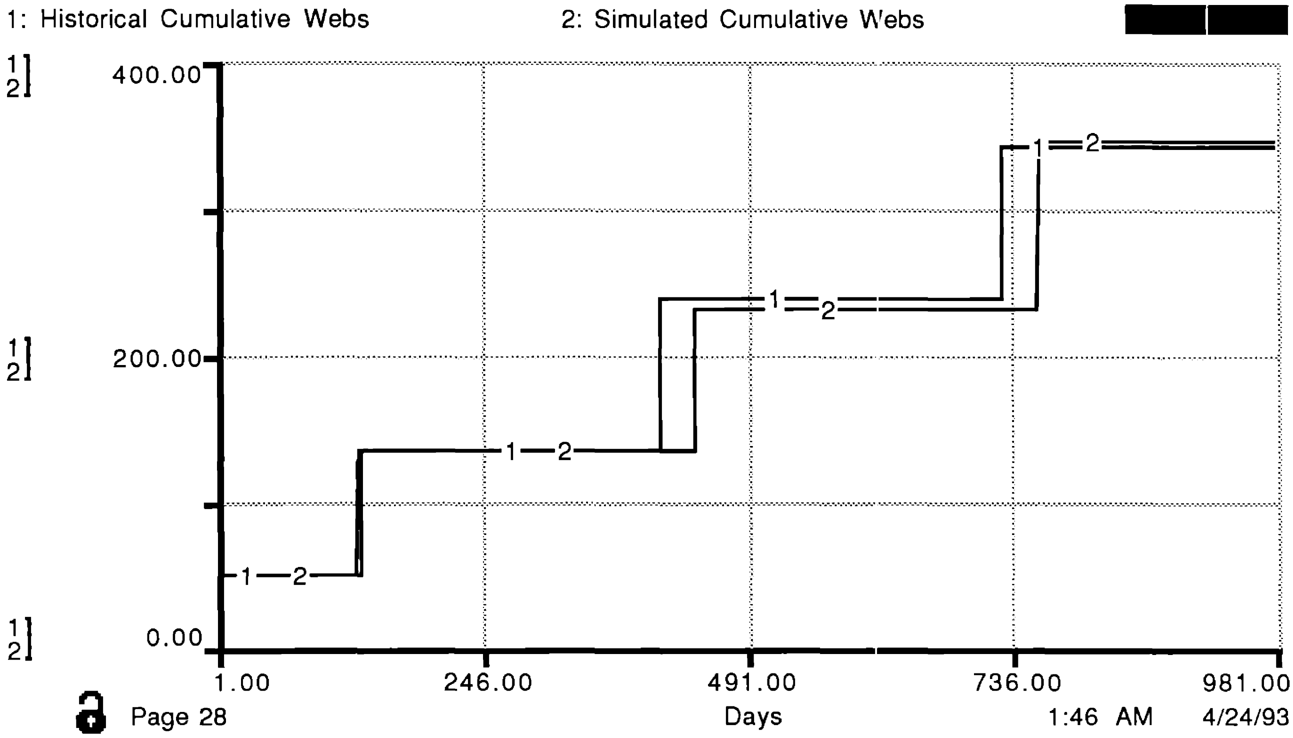


Figure 4.2b - Historical vs. Simulated Cumulative Web Deliveries

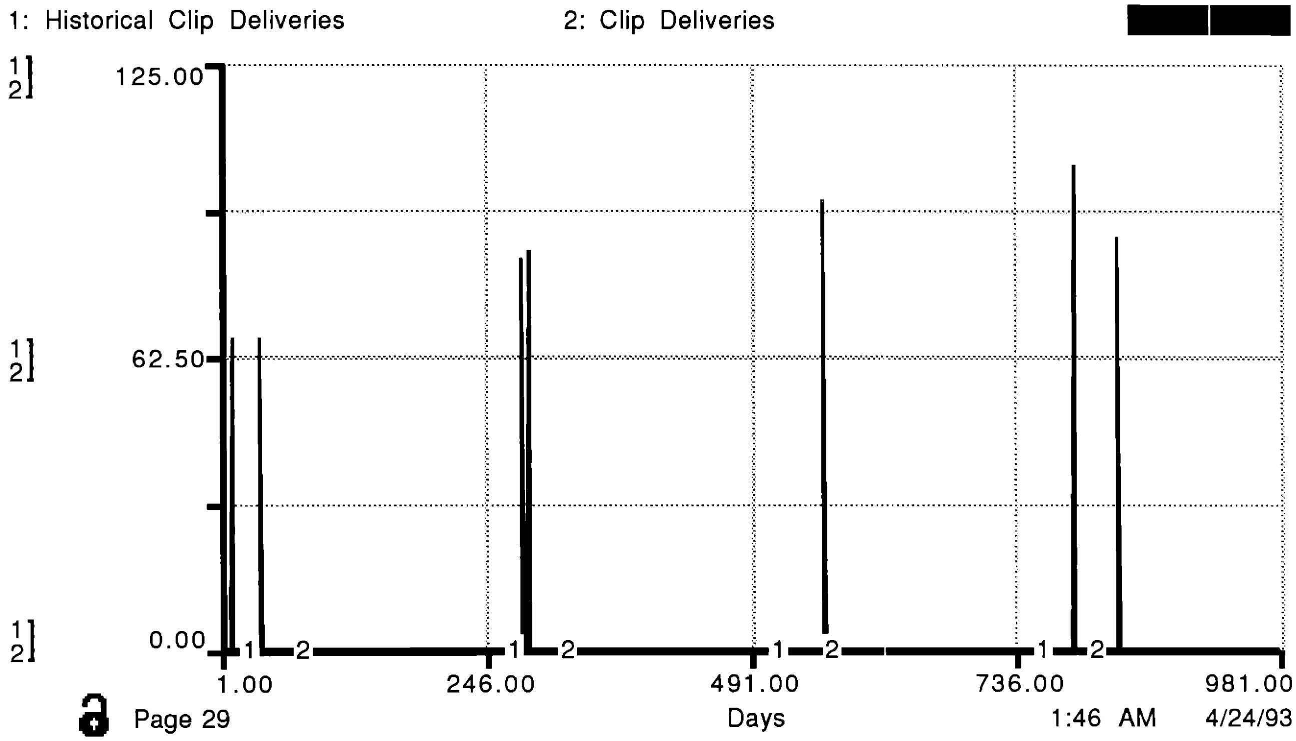


Figure 4.3a - Historical vs. Simulated Clip Deliveries

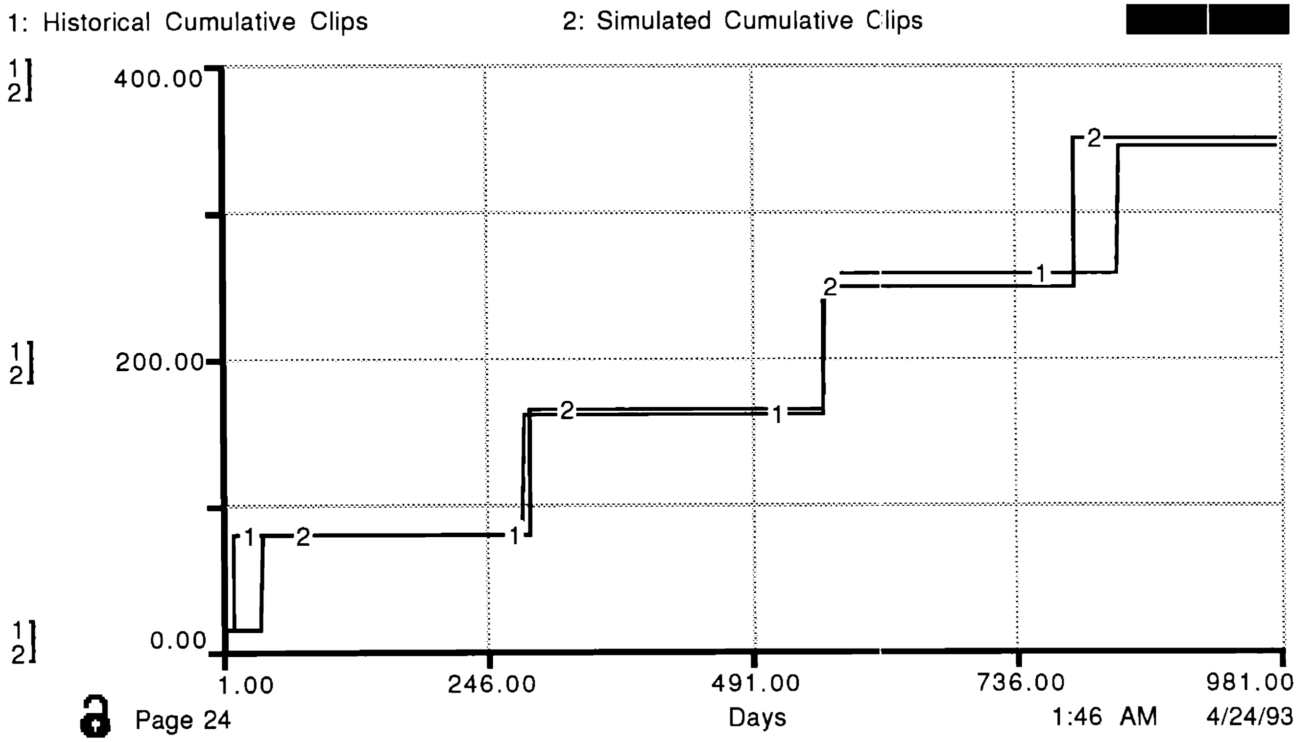


Figure 4.3b - Historical vs. Simulated Cumulative Clip Deliveries

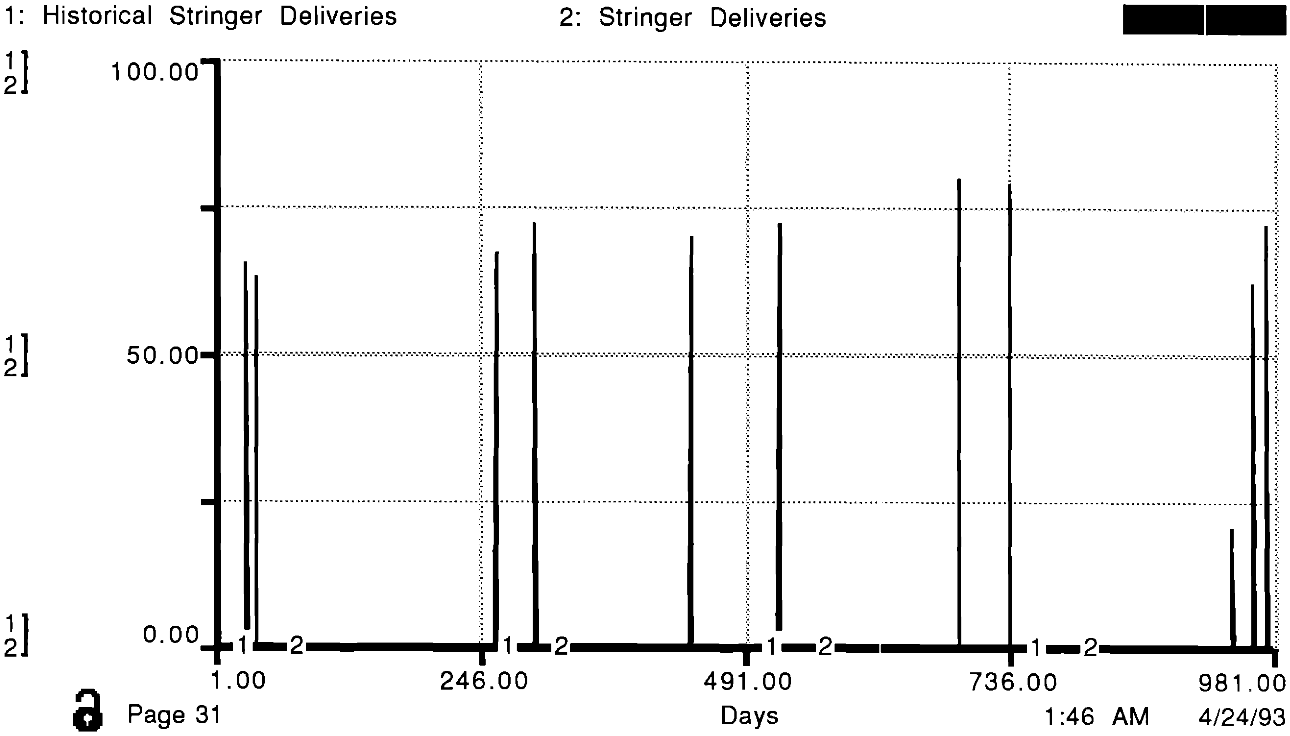


Figure 4.4a - Historical vs. Simulated Stringer Deliveries

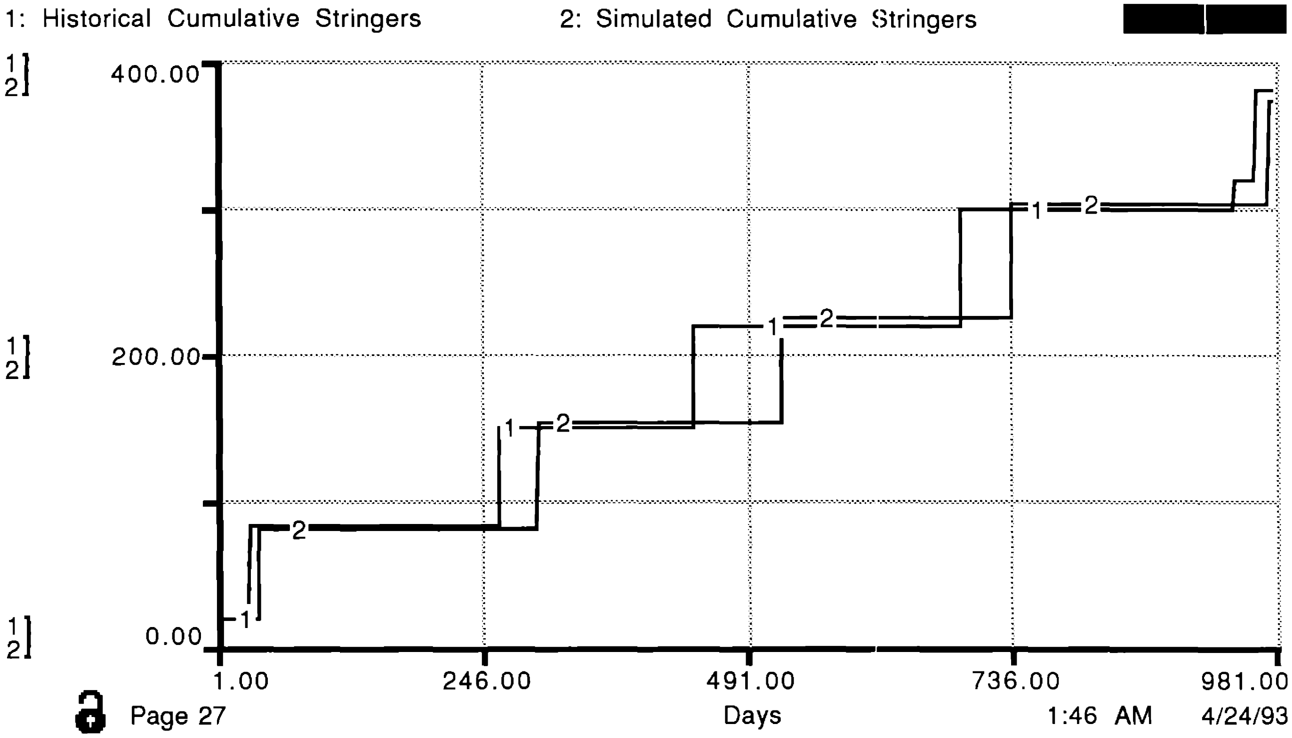


Figure 4.4b - Historical vs. Simulated Cumulative Stringer Deliveries

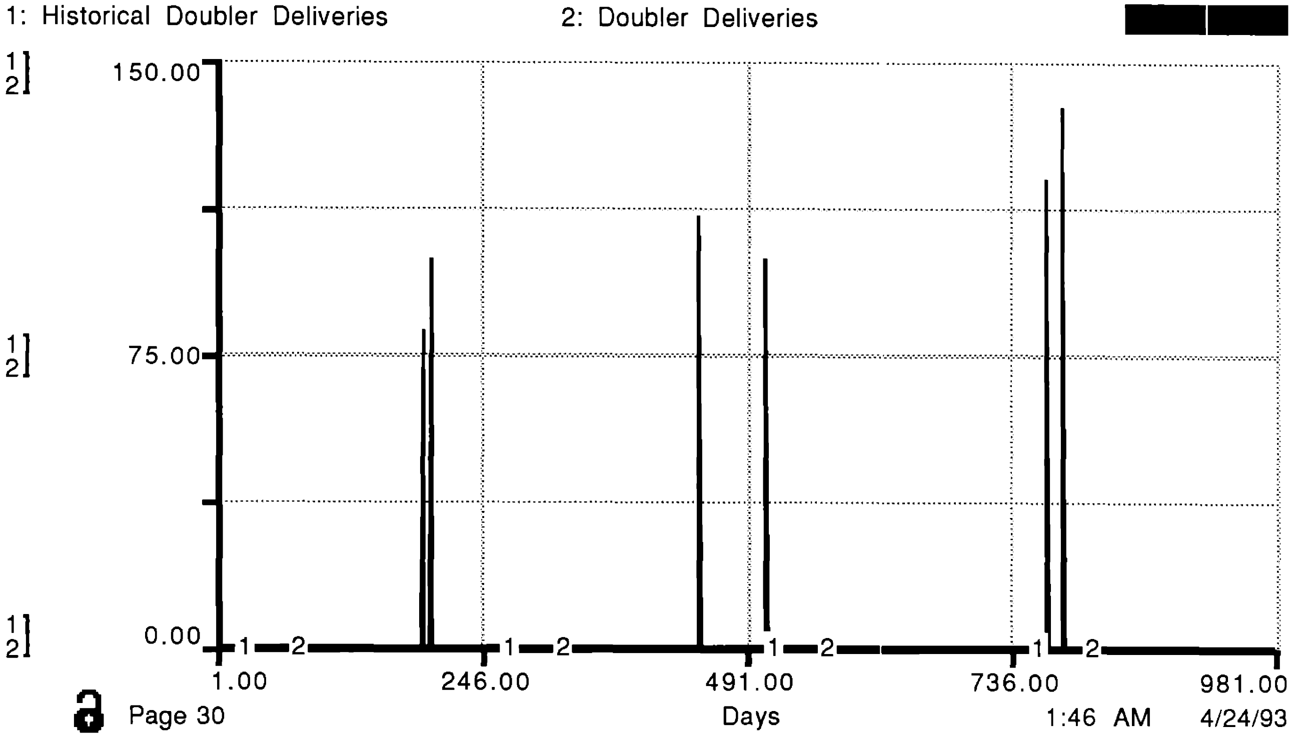


Figure 4.5a - Historical vs. Simulated Doubler Deliveries

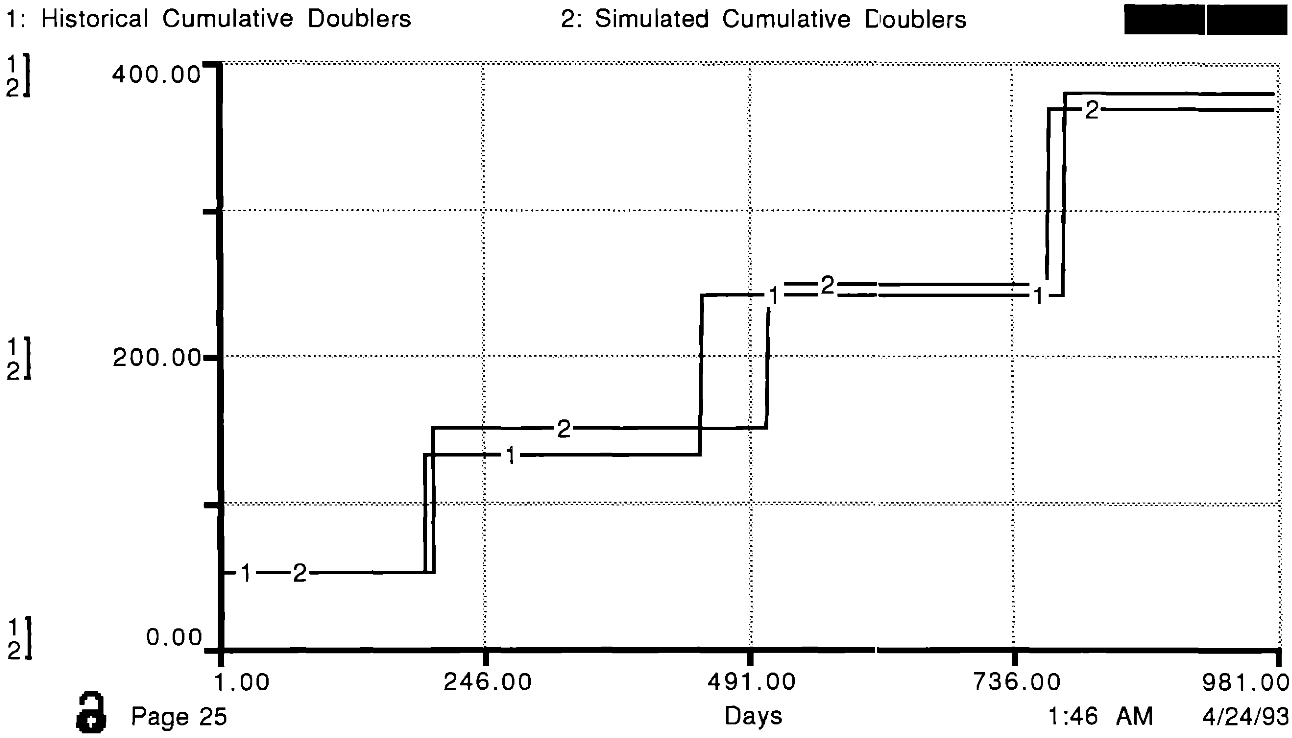


Figure 4.5b - Historical vs. Simulated Cumulative Doubler Deliveries

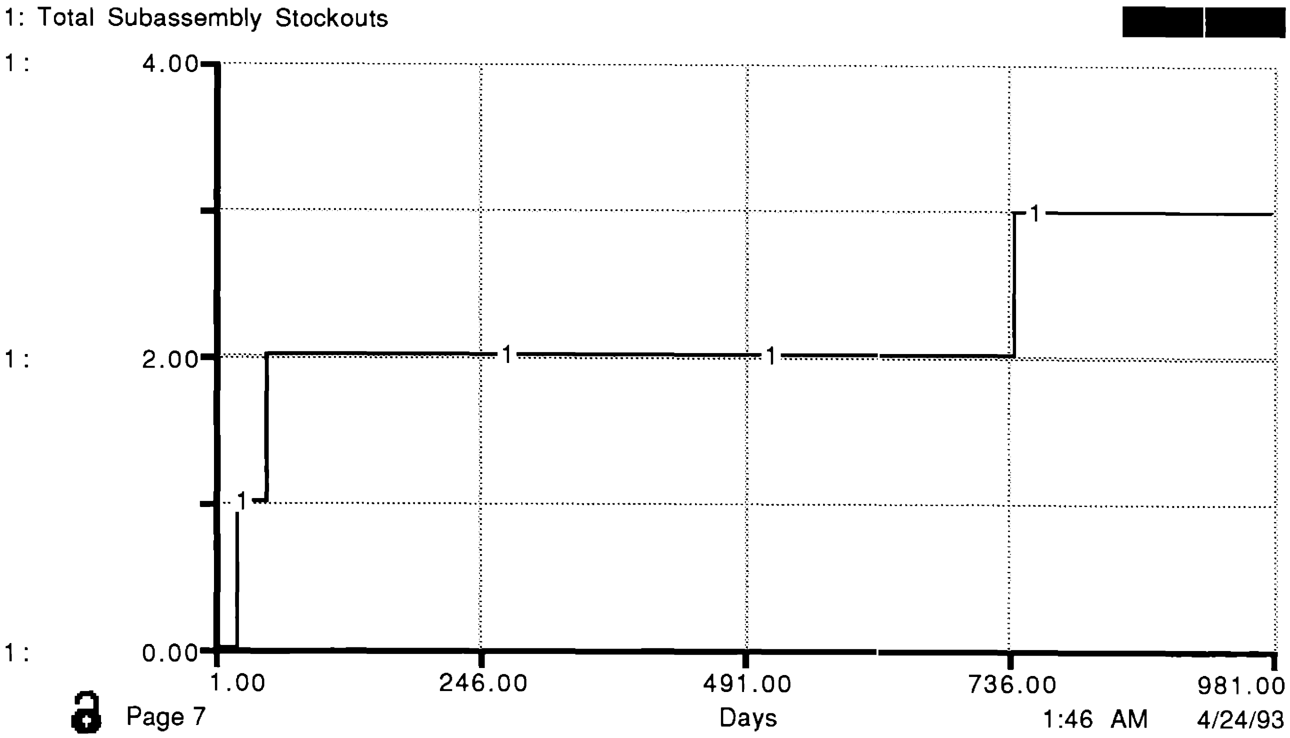


Figure 4.6 - Total Subassembly Stockouts (from Simulation)

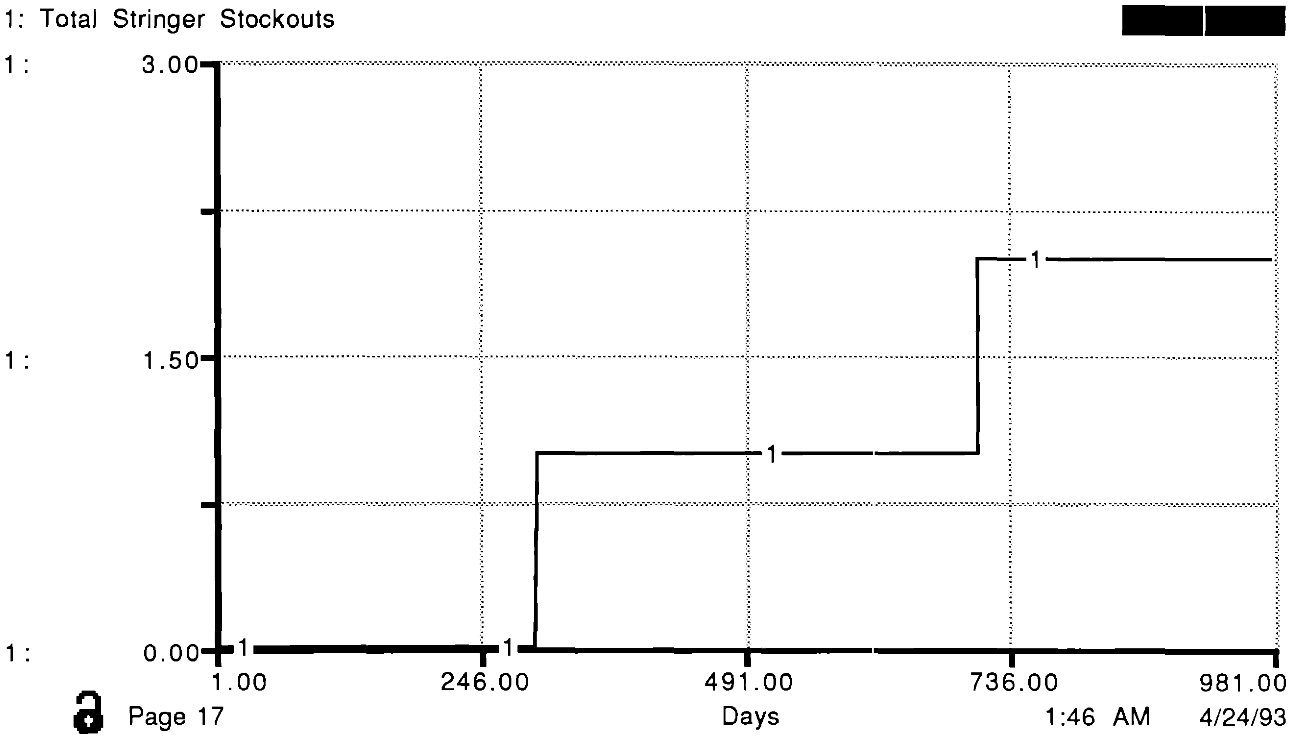


Figure 4.7 - Total Stringer Stockouts (from Simulation)

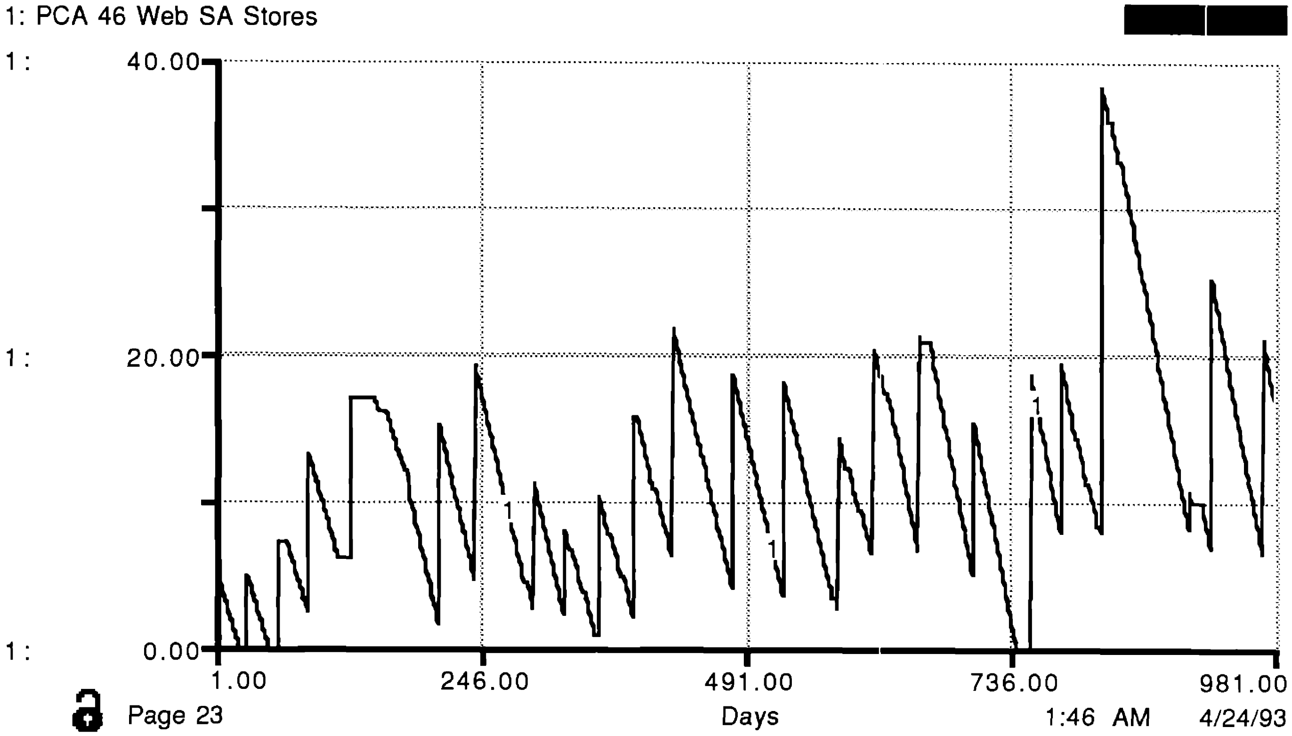


Figure 4.8 - Subassembly Inventory in PCA 46 (from Simulation)

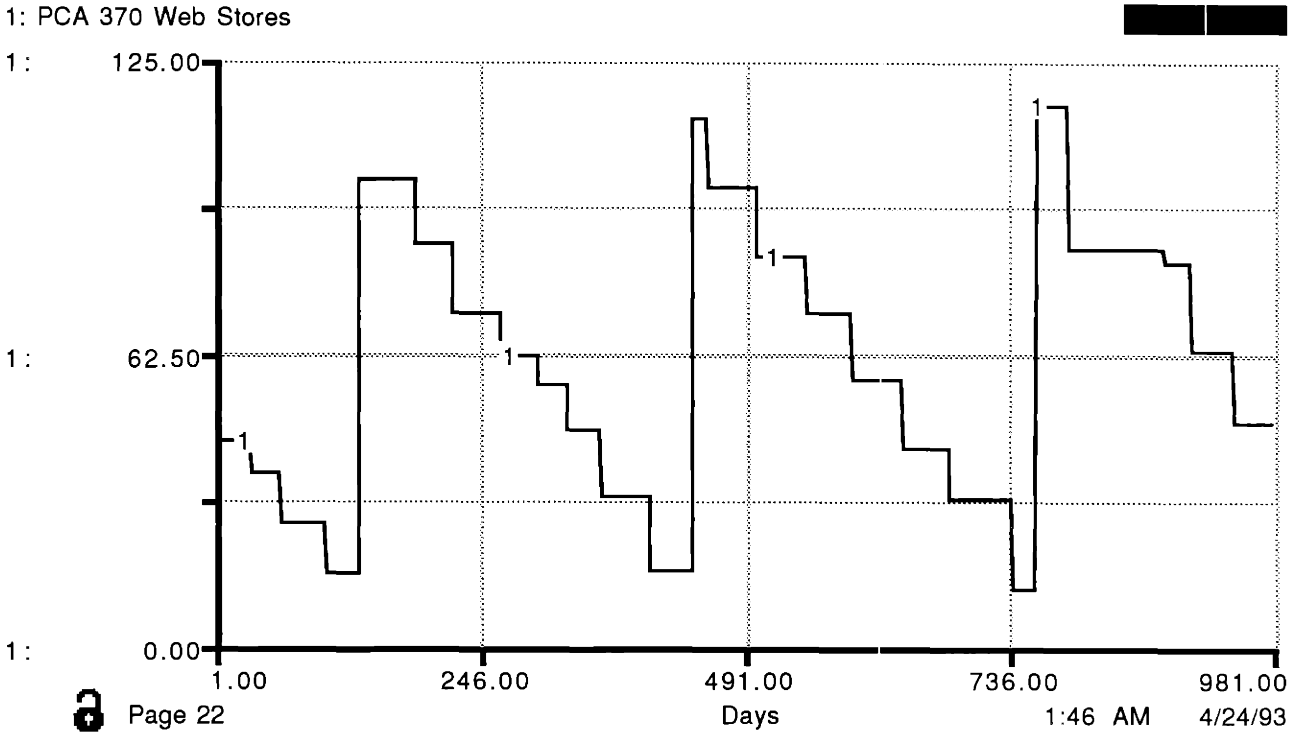


Figure 4.9 - Web Inventory in PCA 370 (from Simulation)

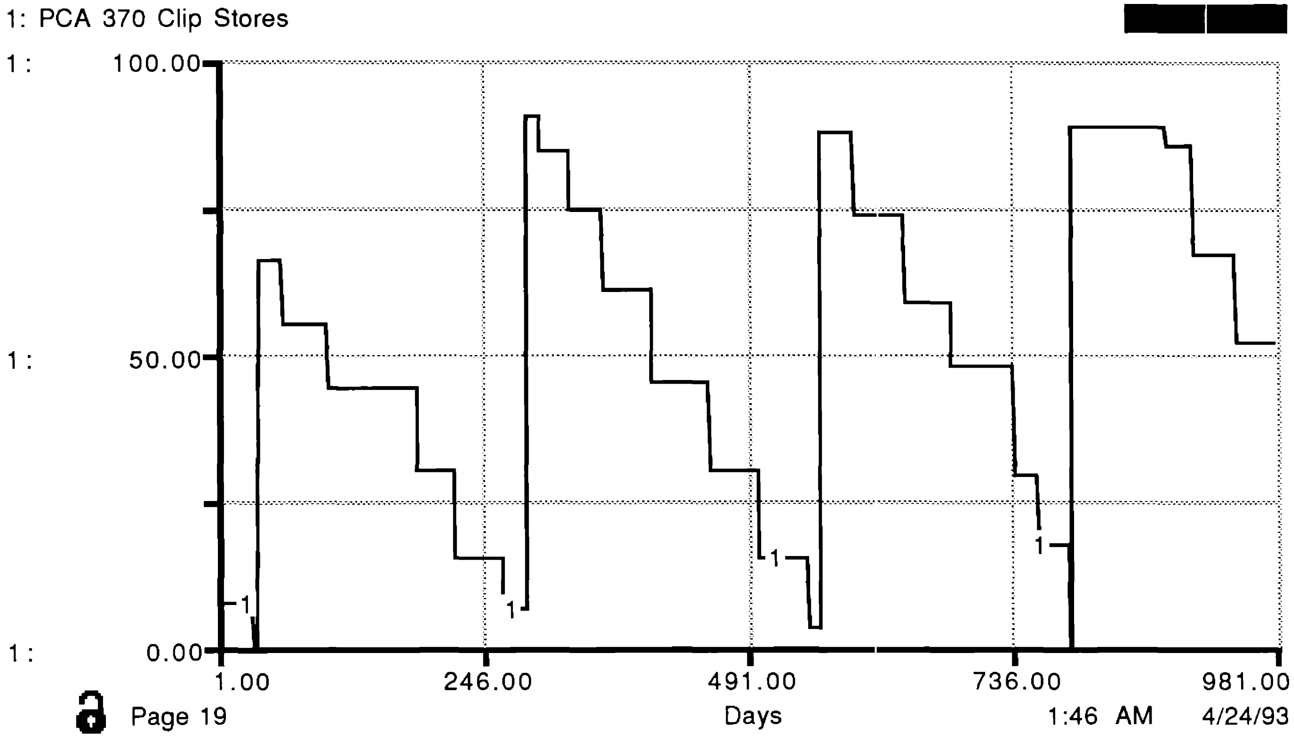


Figure 4.10 - Clip Inventory in PCA 370 (from Simulation)

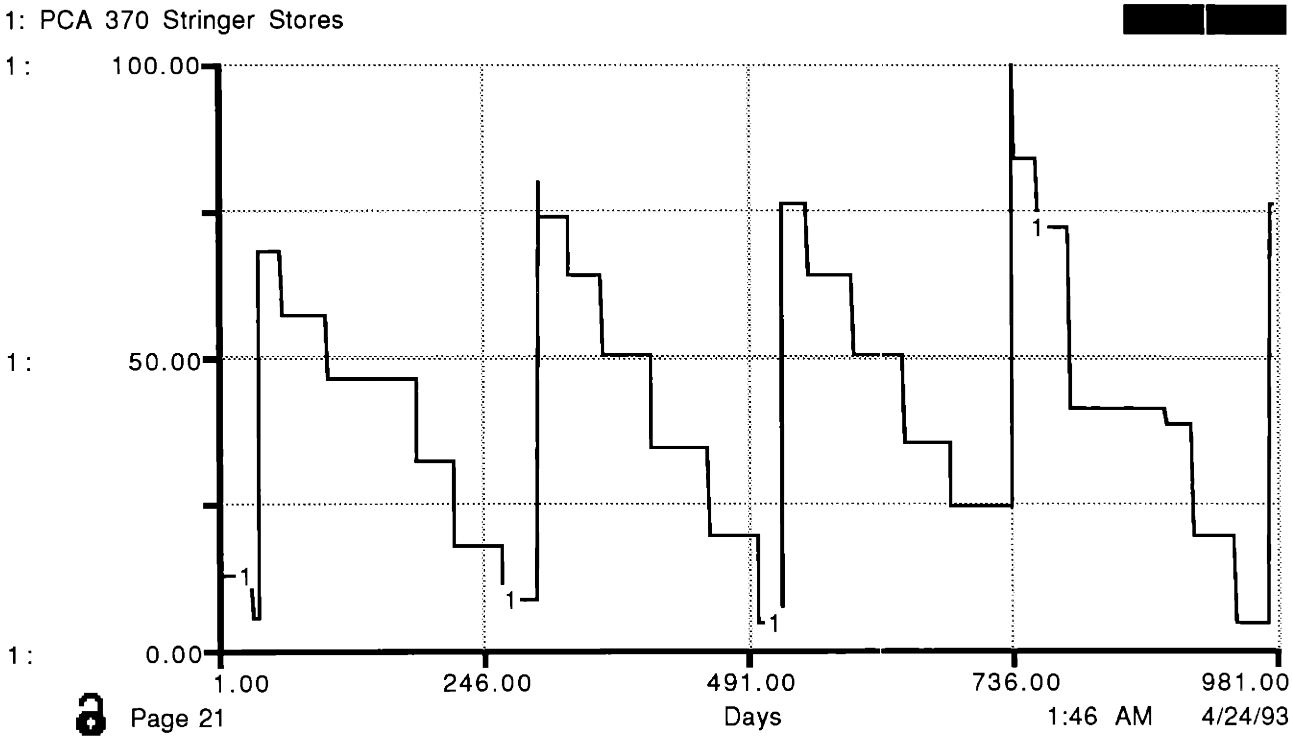


Figure 4.11 - Stringer Inventory in PCA 370 (from Simulation)

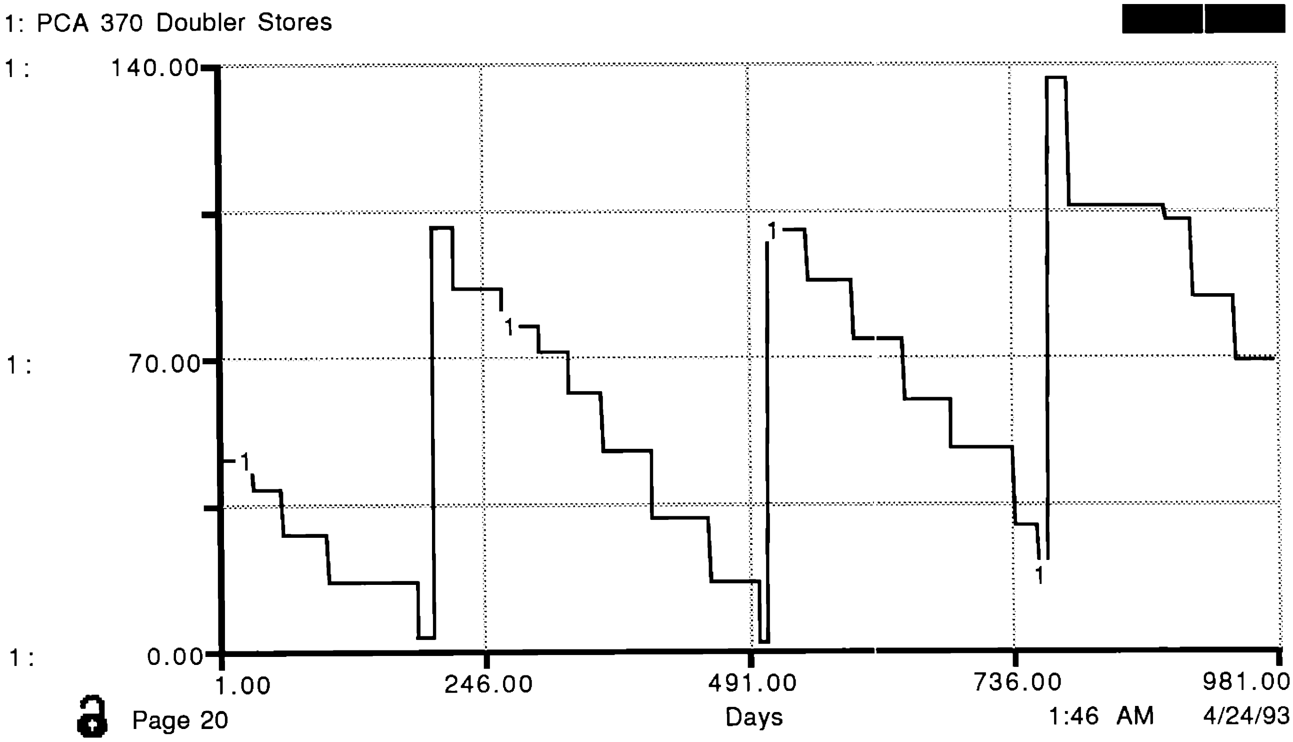


Figure 4.12 - Doubler Inventory in PCA 370 (from Simulation)

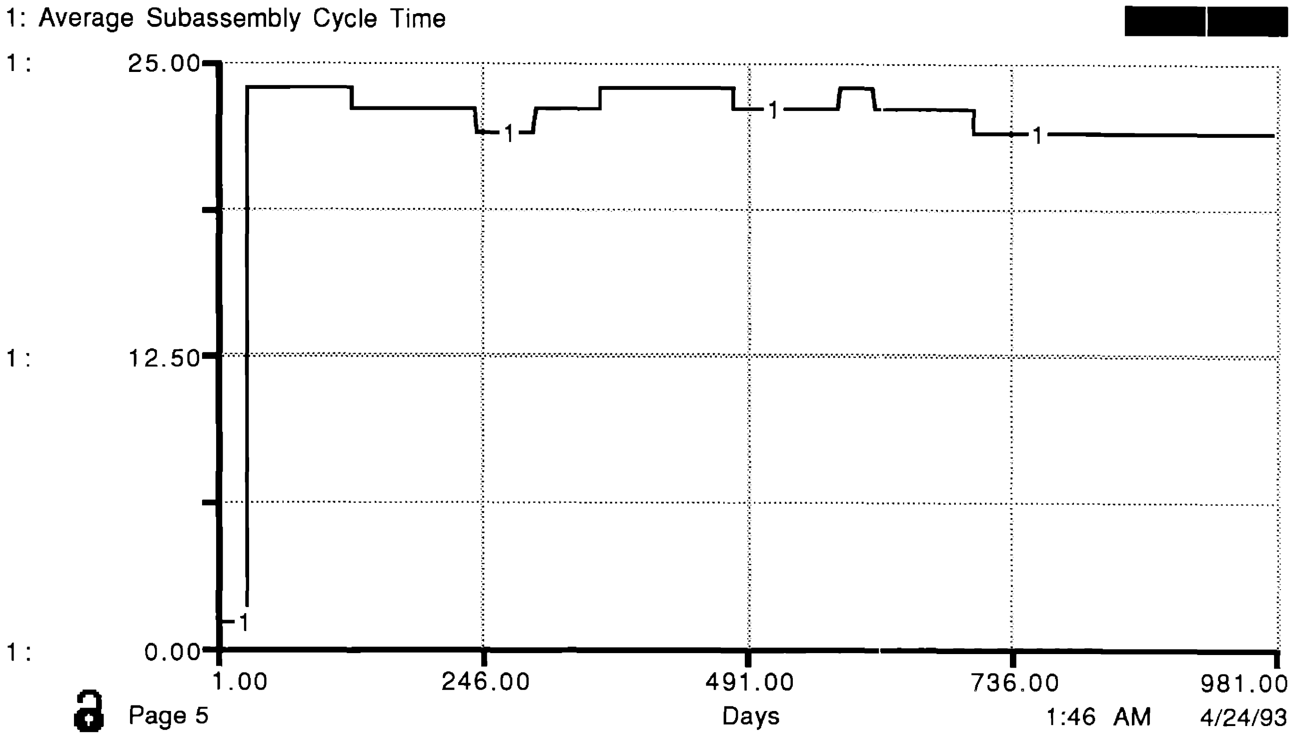


Figure 4.13 - Average Subassembly Cycle Time (from Simulation)

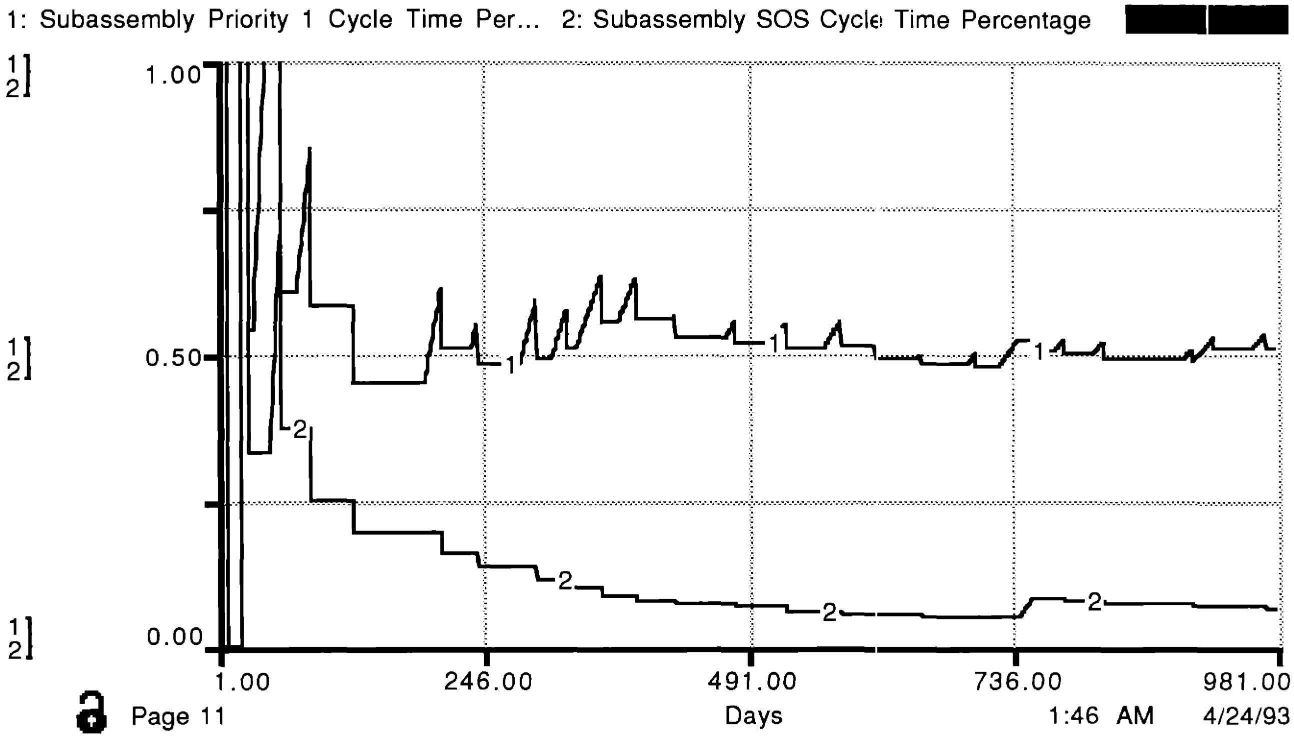


Figure 4.14 - Subassembly Priority Levels (from Simulation)

Chapter 5 - Policy Analysis

The purpose of policy analysis is to use the model as a tool to try out different policy approaches. The model becomes a "managerial microworld" in which the manager can experiment with different policy alternatives. There are two fundamental goals of this analysis: 1) to gain better understanding of the system, and 2) to identify leverage points (areas where change can have the most impact) which can be used to improve system performance.

Policy Analysis Limitations

Before proceeding into a review and discussion of policy analysis simulations, it is important to review some of the limitations of the approach used. While every attempt has been made to develop a model and follow a modeling process that will faithfully generate the behavior of the actual system, time limitations prevent the completion of additional tests and modifications which would provide an even more accurate picture of the behavior of the actual system. These adjustments will have to be part of continuing efforts and research to improve upon the basic model structure and modelling process used in this project.

One limitation of the policy analysis procedure used was the method of trial run data collection. The statistical distributions used will in fact produce the mean values and appropriate variance, but this is more faithfully demonstrated when a very large number of runs are performed all with different random number seeds. The more formal way to check model response with different parameter values would be to perform 20 or more trial runs with the same basic statistical distribution but different random number seeds and then average the results. In this way, any errors due to the random number seed chosen are eliminated. By using only one random number seed for each trial, an easier mechanism is put in place for reproducibility of the tests and their results, but there is some possibility of small errors in trial results.

A limitation of the model, which is relevant to the policy analysis procedure, is the model accounting of expedite and shortage activity. As this structure is currently designed, the model underestimates the impact of expedites and shortages. For example, when the level of subassembly inventory drops below a threshold value, an expediter gets involved to determine what is happening. The model accurately counts this as an expedite action, but stops there.

In actuality, this expeditor may start a chain of expedite actions that have a tremendous perturbative effect on production processing. He or she may increase pressure on Lot Time processing of current orders, split a Lot Time assembly order, call a fellow expeditor in an upstream area to increase pressure on delivery of delinquent detail components or check on the status of detail orders, and call a Materiel Division representative to increase pressure on delivery of externally supplied parts. In essence, there is a multiplier effect of expedite actions that is not accounted for. The reader is cautioned that even in trial runs where expedite and shortage conditions look bad, the situation is usually even much worse because of these multiplier effects.

Initial Run

As a means of comparison with policy tests that will follow, an initial run was performed. A few changes were made from the baseline configuration discussed in the last chapter. First, the initial conditions of the component and subassembly buffer stocks were changed. At the start of all policy runs, the stock levels are full (based on the current ordering policy); starting each simulation from an equilibrium condition aids in the evaluation of different ordering strategies.

Secondly, inventory additions/losses due to shrinkage, quality problems, entry accuracy, overbuild, etc. become more important with attempts to apply the model to the general population of Boeing components. To account for these gains/losses, the specific parameter values listed in Table 5.1 were used. Unless specifically stated, these distributions are active throughout policy testing to model the random effects encountered within the Boeing Ordering System that increase or reduce inventory. What this means is that for every delivery to PCA stores, a percentage of the delivery will be reduced or increased based on these parameter effects.

Table 5.1 - Modified Parameter Values for Policy Analysis

Parameter Description	Distribution Used	
Inventory Additions/Losses	Normal: mean = .03	std dev. = .02
Quality Losses	Normal: mean = 0	std dev. = .02
Entry Accuracy (Record Keeping)	Normal: mean = 1	std dev. = .01

To aid in determining the impact of parameter changes on system performance, a performance metric table was developed. The categories and results from the base run can be seen in Table 5.2. A description of the performance categories is provided in Table 5.3. As different policies are attempted, the impact on these performance metrics will be tracked and will provide a quantitative measure of system behavior.

As an example from the base run, stringer order quantity averaged 60 units. The average inventory level was 34 units. The cycle time for stringers from order release to stringer delivery to PCA 370 averaged 36 days from the base run simulation. Stringer delivery delays caused expediters to get involved on two different occasions and were responsible for two stockout conditions. As different policy approaches are attempted, changes in stringer performance metrics will be compared against the base run. The same approach will be used with the other detail components and the subassembly.

Reducing the Order Quantity

The first policy change calls for a reduction in the time span parameters of all components without an improvement in the underlying process. Essentially this approach calls for a reduction in order quantities independent of manufacturing process capabilities. The reduction in time span parameter by trial number is listed in Table 5.4.

Table 5.2 - Base Run of Policy Simulations

Base Run - Policy Simulations

Performance Metric	**Base Case**
Average Order Quantity (units)	
Stringers	60
Doublers	83
Clips	83
Webs	83
Subassemblies	15
Average Inventory Level (units)	
Stringers	33
Doublers	50
Clips	54
Webs	52
Subassemblies	9
Cycle Time (days)	
Stringers	36
Doublers	40
Clips	169
Webs	179
Subassemblies	23
Number of Expedite Actions (#)	
Stringers	2
Doublers	0
Clips	1
Webs	1
Subassemblies	14
Total	18
Component Stockouts (#)	
Stringers	2
Doublers	0
Clips	1
Webs	1
Total	4
Assembly Stockouts	
Total Number	5
Cumulative Days	14

Table 5.3 - Description of Performance Categories

Average Order Quantity
the average quantity of product ordered over the course of the simulation measured in the number of units per order; this metric is determined by dividing the total amount of product ordered during the simulation by the number of orders for each component or assembly

Average Inventory Level
the average level of inventory held in storage areas over the course of the simulation; this metric is determined by keeping a running total of the inventory level in storage areas and dividing this value by the total simulation time

Average Cycle Time
for the purposes of this analysis the cycle time is defined as the time required for processing of a component or assembly from the start to the conclusion of operations within a work area (it includes queue time as well as touch time); this metric is determined by averaging the cycle times from all orders of a given component type

Number of Expedite Actions
an expedite action occurs whenever an expediter gets involved with the processing of an order; in the model, this occurs when orders are split or within seven days of a stockout condition

Component Stockouts
when detail components are not available in sufficient quantity on the load date of an assembly, a stockout condition has occurred; this metric is determined by counting the number of times that insufficient detail inventory (for the component in question) prevents an assembly order from starting when it was planned to

Assembly Stockouts
when an assembly is not available on the load date of its parent, it is in a stockout condition; the total number is determined by counting all occurrences when the inventory level in the final assembly area drops to zero; the cumulative days represents the total time that the assembly was in a stockout condition over the entire simulation

Table 5.4 - Reducing Order Qty - Order Time Span Changes

Trial #1	Trial #2	Trial #3
20%	35%	40%

The results of the three trials can be seen in Table 5.5 . Inventory levels drop as much as 65-69% for supplier components (i.e. - clips and webs), but a high price is paid. In each trial, the number of expedite actions and stockouts increases. When the order time span is reduced 40%, the supplier deliveries are in constant shortage and require frequent expediter intervention. In addition, the subassembly shortages to final assembly increase from 5 stockouts lasting 14 days to 11 stockouts lasting 48 days.

While it might be tempting to place stress on the system to squeeze out additional inventory reduction, it is clear that pushing the system past its current operational capability can lead to chaos. This policy approach points out the danger of changing ordering policy independent of process capabilities. Reducing order time span coverage should only be reduced in concert with improvement in the manufacturing and distribution processes.

One interesting outcome from the simulation trials is that while supplier delivery performance is adversely effected by a reduction in order quantity, internally produced components do not experience the same problems. In fact, the cycle time of these components actually increases indicating that they are being processed at normal priority levels and not being expedited. This

Table 5.5 - Reduced Order Quantity Performance Chart

Reduced Order Qty		**Policy Alternatives**			**% Change**		
Performance Metric	**Base Case**	**Trial 1 20%**	**Trial 2 35%**	**Trial 3 40%**	**Trial 1 20%**	**Trial 2 35%**	**Trial 3 40%**
Average Order Quantity (units)							
Stringers	60	48	38	37	-20%	-37%	-38%
Doublers	83	69	54	50	-17%	-35%	-40%
Clips	83	69	52	40	-17%	-37%	-52%
Webs	83	69	43	37	-17%	-48%	-55%
Subassemblies	15	12	9	8	-20%	-40%	-47%
Average Inventory Level (units)							
Stringers	33	27	25	24	-18%	-24%	-27%
Doublers	50	37	32	32	-26%	-36%	-36%
Clips	54	43	31	19	-20%	-43%	-65%
Webs	52	41	22	16	-21%	-58%	-69%
Subassemblies	9	8	7	6	-11%	-22%	-33%
Cycle Time (days)							
Stringers	36	36	37	37	0%	3%	3%
Doublers	40	43	43	43	8%	8%	8%
Clips	169	170	159	158	1%	-6%	-7%
Webs	179	185	161	161	3%	-10%	-10%
Subassemblies	23	21	20	20	-9%	-13%	-13%
Number of Expedite Actions (#)							
Stringers	2	1	1	1	-50%	-50%	-50%
Doublers	0	0	0	0	NC	NC	NC
Clips	1	0	2	6	-100%	100%	500%
Webs	1	1	5	7	0%	400%	600%
Subassemblies	14	17	26	28	21%	86%	100%
Total	18	23	34	42			
Component Stockouts (#)							
Stringers	2	1	1	1	-50%	-50%	-50%
Doublers	0	0	0	0	NC	NC	NC
Clips	1	0	4	11	-100%	300%	1000%
Webs	1	0	5	7	-100%	400%	600%
Total	4	2	10	19			
Assembly Stockouts							
Total Number	5	3	6	11	-40%	20%	120%
Cumulative Days	14	5	16	48	-64%	14%	243%

indicates that there may be buffer available within the production environment for those two components. It might be possible to reduce the order quantities without substantial risk of chronic shortages.

Possible Improvement within Existing Process Constraints

Is it possible, within the constraints of current manufacturing process capability, to reduce order sizes without dramatically increasing the level of shortages or expedite action required ? To test this hypothesis, the order time spans were changed for stringers and doublers as shown in Table 5.6. The model results to these changes are listed in Table 5.7.

Table 5.6 - Order Time Span (days)

Item (Base Case)	Trial #1	Trial #2	Trial #3
Stringer (180)	126	84	42
Doubler (252)	126	84	42

As expected, the time span reduction leads to lower inventory levels for both stringers and doublers (a 18-88 % reduction). The first two reductions have a mild impact on expedite actions and shortages, but reducing the time span for ordering to 42 days causes a significant increase in ordering instability (12 expedites for stringers and 18 for doublers). As Trial 2 indicates, there is room for reduction in the inventory levels of stringers and doublers (42% and 64% respectively) with a small penalty in expedite actions and product shortages. Interestingly, subassembly shortages actually decrease during these runs indicating that the possibility exists that smaller lot sizes increase the

Table 5.7 - Improvement within Constraints Performance Chart

Improvement within Constraints		**Policy Alternatives**			**% Change**		
Performance Metric	**Base Case**	**Trial 1 (126)**	**Trial 2 (84)**	**Trial 3 (42)**	**Trial 1 (126)**	**Trial 2 (84)**	**Trial 3 (42)**
Average Order Quantity (units)							
Stringers	60	43	29	15	-28%	-52%	-75%
Doublers	83	41	29	14	-51%	-65%	-83%
Clips	83	83	83	83	0%	0%	0%
Webs	83	83	83	83	0%	0%	0%
Subassemblies	15	14	14	14	-7%	-7%	-7%
Average Inventory Level (units)							
Stringers	33	27	19	9	-18%	-42%	-73%
Doublers	50	29	18	6	-42%	-64%	-88%
Clips	54	51	51	50	-6%	-6%	-7%
Webs	52	45	47	48	-13%	-10%	-8%
Subassemblies	9	9	9	8	0%	0%	-11%
Cycle Time (days)							
Stringers	36	37	35	35	3%	-3%	-3%
Doublers	40	41	43	40	3%	8%	0%
Clips	169	170	170	170	1%	1%	1%
Webs	179	179	172	179	0%	-4%	0%
Subassemblies	23	21	21	22	-9%	-9%	-4%
Number of Expedite Actions (#)							
Stringers	2	5	2	12	150%	0%	500%
Doublers	0	3	2	18	INC	INC	INC
Clips	1	0	0	0	-100%	-100%	-100%
Webs	1	2	1	1	100%	0%	0%
Subassemblies	14	14	13	17	0%	-7%	21%
Total	18	24	18	48			
Component Stockouts (#)							
Stringers	2	2	1	4	0%	-50%	100%
Doublers	0	2	0	6	INC	NC	INC
Clips	1	0	0	0	-100%	-100%	-100%
Webs	1	1	1	1	0%	0%	0%
Total	4	5	2	11			
Assembly Stockouts							
Total Number	5	3	3	5	-40%	-40%	0%
Cumulative Days	14	11	12	12	-21%	-14%	-14%

flexibility of assembly area to respond to system variation. The reason that stockouts do not increase with expedites in Trial 3 is because while stringer and doubler processing capabilities are stretched, the priority and expedite system is capable of delivering the components prior to a shortage condition, in most cases. However, stockout performance degrades rapidly if the time span is reduced too far below 42 days (based on additional runs not shown).

One limitation of this analysis is that the model characterizes the underlying performance of the manufacturing segments based on historical data. By reducing the order sizes by a factor of three (from 252 day coverage to 84 days), the underlying process distribution is likely to shift. Until Boeing improves plant layout, reduces set-up time, utilizes group technology, and develops other lean manufacturing techniques, the mean time required to process an order will likely increase. So it pays to be careful when making adjustments to order sizes. The point of this policy test was not to demonstrate the optimum level of detail and assembly order sizes, but rather to indicate that there might be room within the performance constraints of the current system to reduce order size without jeopardizing delivery performance. Is the same approach applicable with external suppliers ?

Reducing Supplier Reorder Lead Times (ROLTs)

Currently Boeing suppliers require nine months of lead time on average (see Fig. 2.5) prior to delivery of requested orders. Boeing has been working with its suppliers to improve their quality (i.e. - Advanced Quality System (AQS)), and they have also recently started a program to rationalize their supplier base. In this new environment it is feasible that supplier ROLTs could be significantly reduced. The actual fabrication time for most Purchased Outside

Production (POP) parts is measured in days, if not hours. Based on discussions with representatives from Boeing's Materiel Division, ROLTs on the order of 60 days or less are quite possible. In this policy test, it is assumed that supplier ROLTs are reduced from 180 days to 60 days, and order time spans are reduced as identified in Table 5.8.

Table 5.8 - Order Time Span (days)

Component	Trial #1	Trial #2	Trial #3
Webs	84	63	42
Clips	84	63	42

The results of this test are presented in Table 5.9, and as expected, the response of the system shows a reduction in inventory coverage (54-65% thru first two trials) for externally supplied parts. The effect on process stability is mild until trial #3. With time span coverage reduced to 42 days the system is unable to meet demand in final assembly, and therefore there are 18 stockouts lasting 122 days. While reduction of supplier ROLTs has obvious benefits for inventory reduction, once the inventory coverage drops below supplier process capability shortages are common. An interesting side effect in Trial 3 is the increase in stringer and doubler inventory levels (33 & 30%). This increase is a result of the chronic shortages of supplier parts; subassembly kits can not be filled until the delinquent parts arrive. As a result, doublers and stringers spend more time in storage increasing the average inventory level for those parts.

Table 5.9 - ROLT Reduction Performance Chart

ROLT Reduction		**Policy Alternatives**			**% Change**		
Performance Metric	**Base Case**	**Trial 1 (84)**	**Trial 2 (63)**	**Trial 3 (42)**	**Trial 1 (84)**	**Trial 2 (63)**	**Trial 3 (42)**
Average Order Quantity (units)							
Stringers	60	60	60	60	0%	0%	0%
Doublers	83	83	83	83	0%	0%	0%
Clips	83	29	22	14	-65%	-73%	-83%
Webs	83	29	22	10	-65%	-73%	-88%
Subassemblies	15	15	15	8	0%	0%	-47%
Average Inventory Level (units)							
Stringers	33	34	34	44	3%	3%	33%
Doublers	50	50	50	65	0%	0%	30%
Clips	54	25	19	14	-54%	-65%	-74%
Webs	52	22	18	3	-58%	-65%	-94%
Subassemblies	9	9	9	5	0%	0%	-44%
Cycle Time (days)							
Stringers	36	36	36	37	0%	0%	3%
Doublers	40	40	40	42	0%	0%	5%
Clips	169	53	52	46	-69%	-69%	-73%
Webs	179	57	56	51	-68%	-69%	-72%
Subassemblies	23	23	23	20	0%	0%	-13%
Number of Expedite Actions (#)							
Stringers	2	3	2	1	50%	0%	-50%
Doublers	0	0	0	0	NC	NC	NC
Clips	1	0	1	14	-100%	0%	1300%
Webs	1	1	5	29	0%	400%	2800%
Subassemblies	14	13	14	22	-7%	0%	57%
Total	18	17	22	66			
Component Stockouts (#)							
Stringers	2	2	1	1	0%	-50%	-50%
Doublers	0	0	0	0	INC	INC	INC
Clips	1	0	0	18	-100%	-100%	1700%
Webs	1	1	2	31	0%	100%	3000%
Total	4	3	3	50			
Assembly Stockouts							
Total Number	5	5	4	18	0%	-20%	260%
Cumulative Days	14	14	18	122	0%	29%	771%

What's in the bins ?

With literally thousands of parts in storage at Boeing Production Control Areas (PCAs), inventory accuracy is a difficult task. Inventory is lost due to shrinkage, spares use, or AOG requirements, and in some cases excess inventory resides in the bins because of overbuild from work areas. For this simulation, the inventory loss distribution shown in Table 5.10 was used. The model implements this by losing/gaining a percentage of every shipment to PCA 370, but the ordering system logic is told that the entire shipment was delivered as requested. Typically the ordering system will be updated when either a shortage[1] occurs or an inventory audit is performed. The results of this test can be seen in Table 5.11. The average inventory level decreases for most components. Stringer inventory level increases for the first two trials. It should be remembered that the inventory loss is stochastic. It is quite possible, as in the case of the stringer, that there will not be a loss when an order arrives. As the mean level of inventory increases the stockout duration

Table 5.10 - Inventory Loss Distribution

Component	Trial #1	Trial #2	Trial #3
Mean	.05	.07	.09
Std Dev	.02	.02	.02

---

[1] Model assumption.

Table 5.11 - Inventory Loss Performance Chart

Inventory Accuracy		**Policy Alternatives**			**% Change**		
Performance Metric	**Base Case**	**Trial 1** $\mu = .05$	**Trial 2** $\mu = .07$	**Trial 3** $\mu = .09$	**Trial 1** $\mu = .05$	**Trial 2** $\mu = .07$	**Trial 3** $\mu = .09$
Average Order Quantity (units)							
Stringers	60	60	60	62	0%	0%	3%
Doublers	83	78	83	83	-6%	0%	0%
Clips	83	83	79	82	0%	-5%	-1%
Webs	83	82	82	82	-1%	-1%	-1%
Subassemblies	15	14	14	15	-7%	-7%	0%
Average Inventory Level (units)							
Stringers	33	35	34	30	6%	3%	-9%
Doublers	50	48	46	44	-4%	-8%	-12%
Clips	54	48	50	50	-11%	-7%	-7%
Webs	52	46	47	44	-12%	-10%	-15%
Subassemblies	9	8	8	8	-11%	-11%	-11%
Cycle Time (days)							
Stringers	36	37	34	37	3%	-6%	3%
Doublers	40	39	40	42	-3%	0%	5%
Clips	169	172	170	174	2%	1%	3%
Webs	179	182	175	181	2%	-2%	1%
Subassemblies	23	23	21	21	0%	-9%	-9%
Number of Expedite Actions (#)							
Stringers	2	4	2	1	100%	0%	-50%
Doublers	0	1	3	3	INC	INC	INC
Clips	1	2	2	1	100%	100%	0%
Webs	1	2	1	2	100%	0%	100%
Subassemblies	14	20	19	19	43%	36%	36%
Total	18	29	27	26			
Component Stockouts (#)							
Stringers	2	2	2	1	0%	0%	-50%
Doublers	0	1	2	1	INC	INC	INC
Clips	1	2	2	2	100%	100%	100%
Webs	1	1	1	1	0%	0%	0%
Total	4	6	7	5			
Assembly Stockouts							
Total Number	5	5	6	8	0%	20%	60%
Cumulative Days	14	21	32	32	50%	129%	129%

increases accordingly. The major problem with inventory loss of this type is the delay in discovery. When the loss is uncovered, a shortage is almost guaranteed.

Product Quality

Component rejections due to poor workmanship and engineering changes continue to place strains on Boeing manufacturing areas leading to component shortages. The large lot sizes used when ordering compound the problem. When a lot is rejected, it usually means that a year's worth of product has to be scrapped or reworked.

This policy test evaluates the system response to various levels of product loss due to quality problems. The product quality loss distribution used for the three trials can be seen in Table 5.12.

Table 5.12 - Product Quality Loss Distribution

Component	Trial #1	Trial #2	Trial #3
Mean	.10	.20	.30
Std Dev	.02	.02	.02

The results of these trials are shown in Table 5.13. The number of expedites and shortages continues to climb as more product is lost. This instability leads to lower inventory levels and shorter cycle times as the orders are hand carried from one work area to the next by expediters. In Trial 3, with a quality loss of 30%, a total of 14 stockouts lasting 108 days occurs. Certainly Boeing is not going to wait for delinquent assemblies when building an aircraft. The

Table 5.13 - Product Quality Performance Chart

Product Quality		Policy Alternatives			% Change		
Performance Metric	**Base Case**	**Trial 1 μ = .10**	**Trial 2 μ = .20**	**Trial 3 μ = .30**	**Trial 1 μ = .10**	**Trial 2 μ = .20**	**Trial 3 μ = .30**
Average Order Quantity (units)							
Stringers	60	57	59	59	-5%	-2%	-2%
Doublers	83	82	79	79	-1%	-5%	-5%
Clips	83	82	79	62	-1%	-5%	-25%
Webs	83	82	68	61	-1%	-18%	-27%
Subassemblies	15	14	14	12	-7%	-7%	-20%
Average Inventory Level (units)							
Stringers	33	30	30	28	-9%	-9%	-15%
Doublers	50	45	39	39	-10%	-22%	-22%
Clips	54	47	39	24	-13%	-28%	-56%
Webs	52	40	28	21	-23%	-46%	-60%
Subassemblies	9	8	7	5	-11%	-22%	-44%
Cycle Time (days)							
Stringers	36	35	34	34	-3%	-6%	-6%
Doublers	40	40	38	38	0%	-5%	-5%
Clips	169	161	157	152	-5%	-7%	-10%
Webs	179	167	164	160	-7%	-8%	-11%
Subassemblies	23	21	22	21	-9%	-4%	-9%
Number of Expedite Actions (#)							
Stringers	2	4	5	7	100%	150%	250%
Doublers	0	2	3	5	INC	INC	INC
Clips	1	1	4	10	0%	300%	900%
Webs	1	3	7	14	200%	600%	1300%
Subassemblies	14	18	22	27	29%	57%	93%
Total	18	28	41	63			
Component Stockouts (#)							
Stringers	2	2	3	4	0%	50%	100%
Doublers	0	1	2	5	INC	INC	INC
Clips	1	2	4	18	100%	300%	1700%
Webs	1	2	5	11	100%	400%	1000%
Total	4	7	14	38			
Assembly Stockouts							
Total Number	5	5	7	14	0%	40%	180%
Cumulative Days	14	18	21	108	29%	50%	671%

work will become a "green line" which means that it will travel to a downstream control code where it will eventually be completed. The cost and disruption of this work around is staggering, but it is also a fact of life in Boeing final assembly areas.

In Figure 5.1 and 5.2, the system response to a lot of stringers that is scrapped can be seen. In this run a lot of stringers is rejected on simulation day 420 shortly after they arrive in PCA 370. The ordering system learns of the problem and generates a rush order to correct the shortage. The rejected lot's need date drives the rush order which means when this order is released to the shop it is late; not only is it late, it is SOS ! Because the lot of stringers in this example is already SOS, the expediters rush 25% of the order through processing (the first blip), and the remaining portion of the order follows later.

While component rework and loss due to rejection is a cost that all firms want to avoid, the disruption to the normal process flow is an unseen[2] cost. Fig 5.3 provides a clear example of the type of problems that component rejections can cause. This figure displays the distribution of order releases in the Sheet Metal Center for a one year period. As an example, 9,667 orders were released 10 to 30 days late[3] during this time period. Based on this data,

---

[2] The problem is actually very visible when you walk through the factory because every order is late, but the cause is not apparent.

[3] This means that the order arrives after its planned start date. For example, if the need date for a lot of components is M-day 400, and the flowtime allocated for this process is 50 days, the lot should start processing on M-day 350. If the order which requests the lot to be built shows up on M-day 370, the order release is 20 days late.

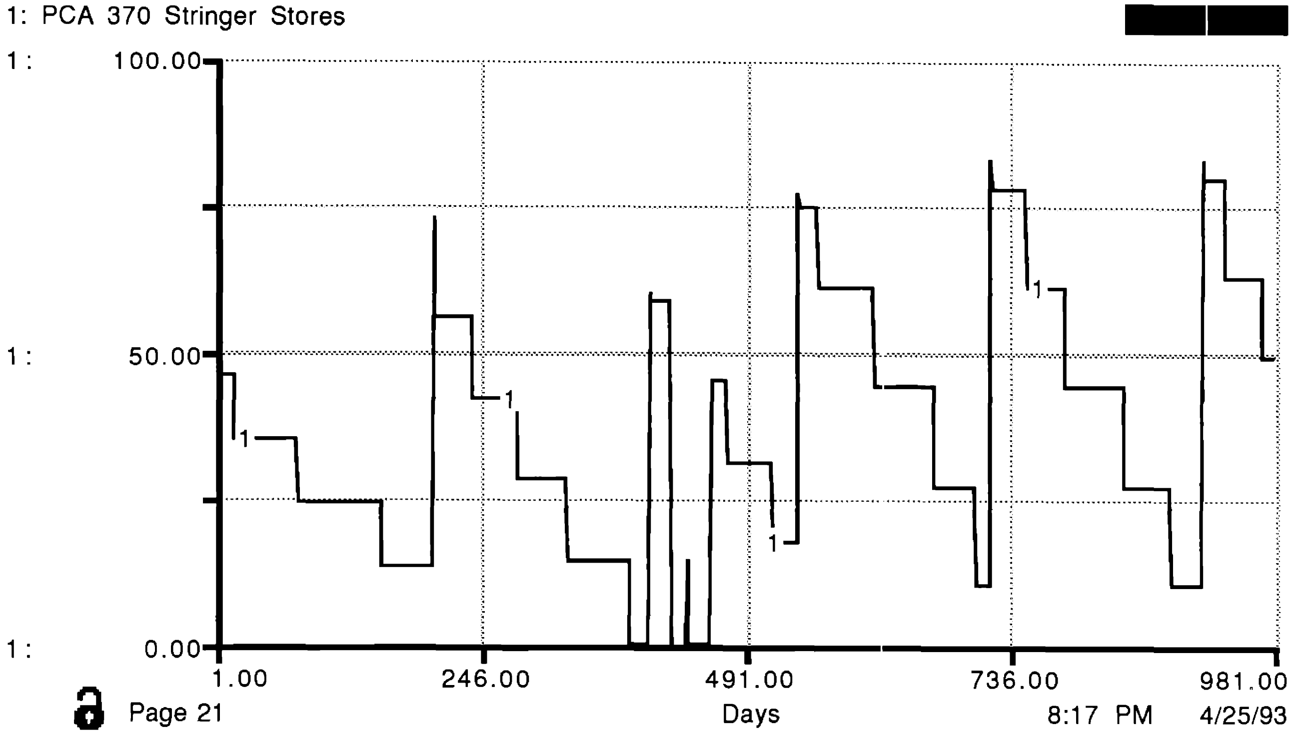


Figure 5.1 - Stringer Stores (Lot Rejection @ t=420)

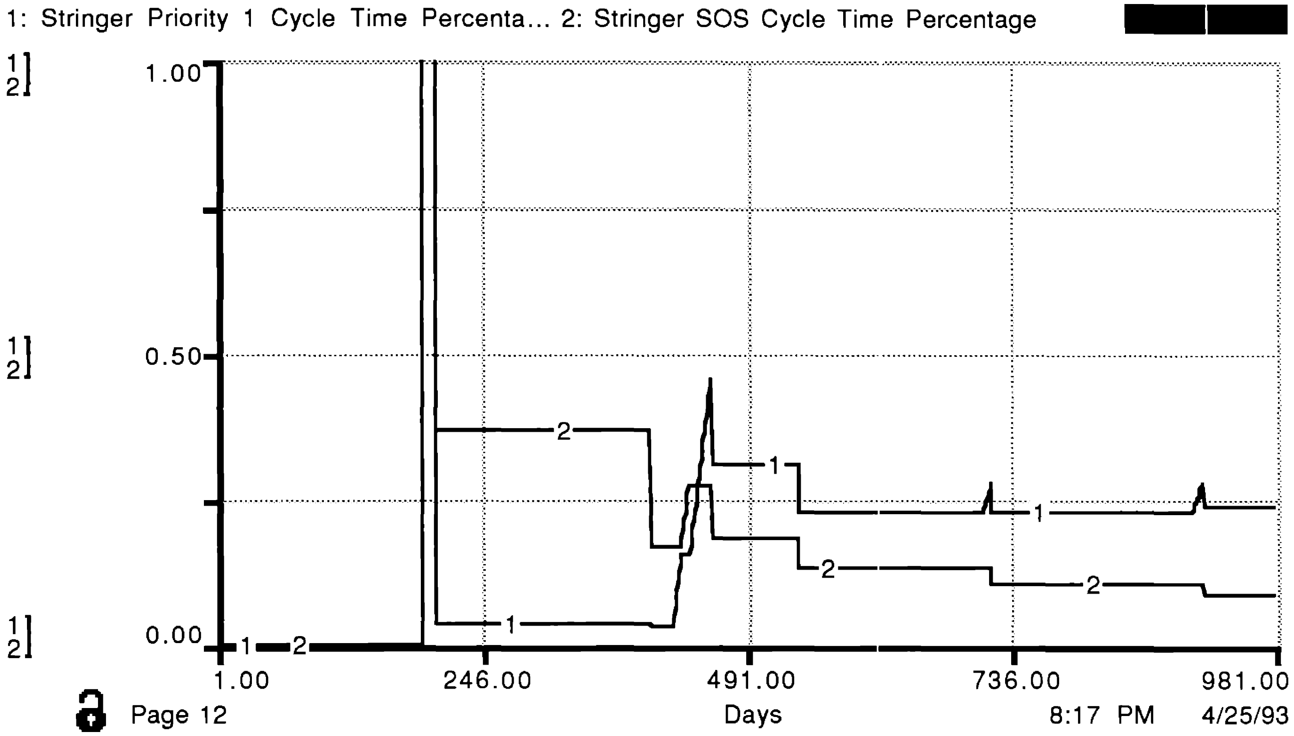
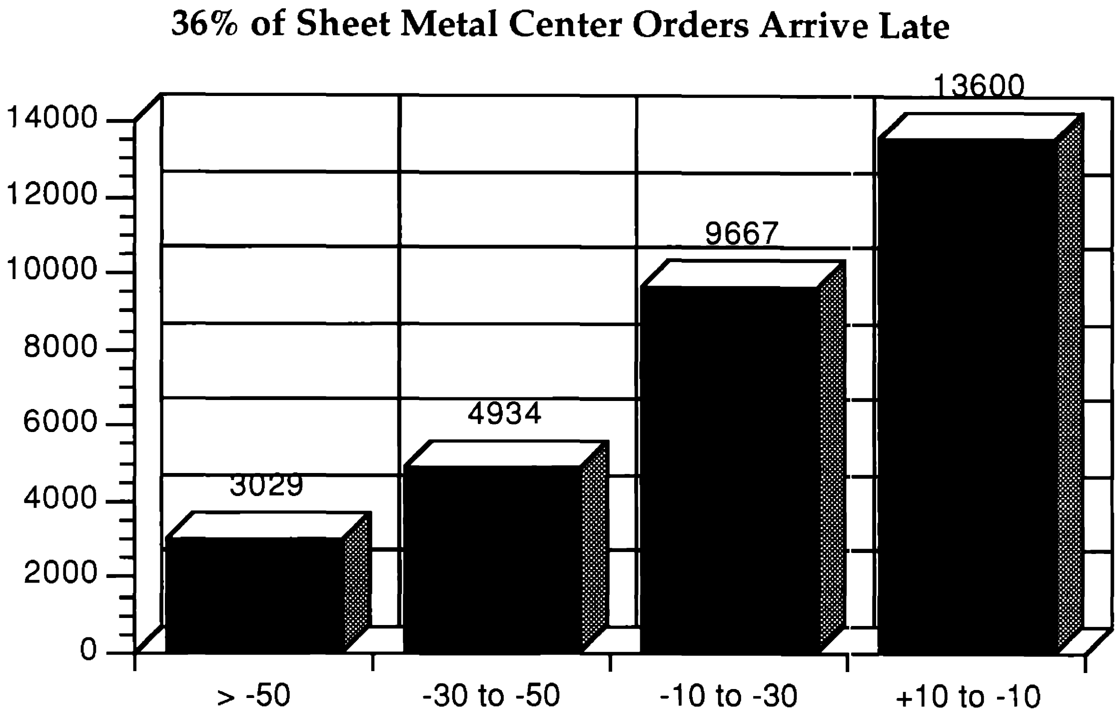


Figure 5.2 - Stringer Priority Level (Lot Rejection @ t=420)



36% of Sheet Metal Center Orders Arrive Late

Figure 5.3 - Sheet Metal Center Order Release Performance

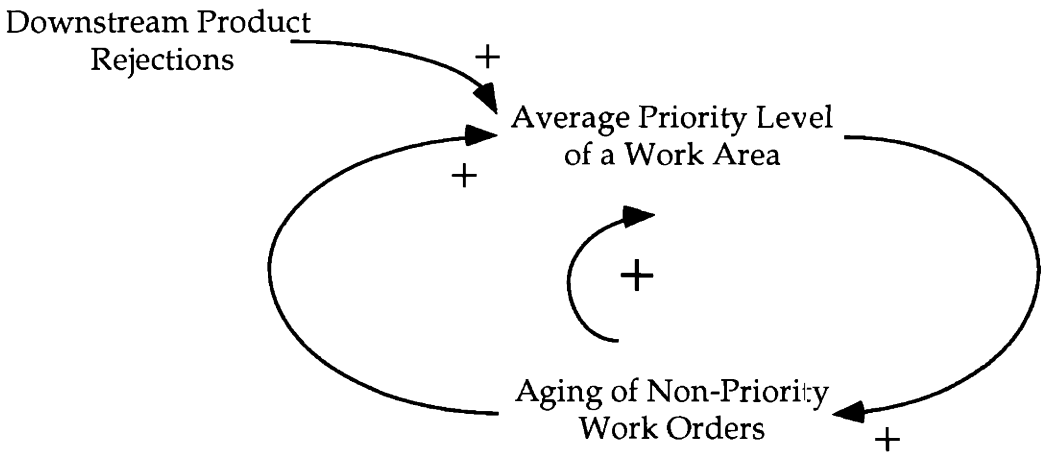


Figure 5.4 - Sheet Metal Priority Cycle

36% of the total number of orders released to the Sheet Metal Center are late when they arrive. What causes this to happen ? Poor product quality rings in as one of the primary root causes, and the simulation provides an example of how this can occur. These type of occurrences lead to another problem dynamic described in Fig. 5.4. An increase in late releases means there will be a corresponding increase in the average priority level of the work area. As the "hot" orders are worked first, non-priority work sits in the queue until it ages and increases in priority. The average priority level of the work area increases. Soon the only work orders that get attention are the ones being expedited or the ones with a very high priority. As long as the percentage of component rejections remains high, it will be very difficult to break the high priority cycle that exists in many Boeing work areas.

Reducing Planning Flowtimes

Puget Sound Flow Standards determine when orders will be released to the production area. For the assembly of components in Lot Time areas, ten days of flowtime are allocated with a twenty day FTBO, and therefore, a total of thirty days of flowtime are planned for. What happens within the system if flowtimes are changed ? In Table 5.14, the subassembly flowtimes used for this test can be seen. The system response can be seen in Table 5.15.

Table 5.14 - Subassembly Flowtime Values (days)

Component	Trial #1	Trial #2
Subassembly	20	40

In Trial #1 the response is similar to reducing the order size to a level that the underlying process finds difficult to support. Subassembly shortages, with frequent expediting, become a continuous fact of life. The average level of inventory for all components increases, but subassembly inventory levels decrease. The reason for this is that subassembly orders are starting later (at 20 days prior to the need date vice 30), and therefore components remain in storage longer. Subassembly stores, on the other hand, are depleted because actual processing in the Lot Time area takes longer than the planning flowtime.

The lesson is simple. If you plan for less time than it really takes, there will be frequent product shortages. What if the planning time is greater than is really needed ? In Trial #2, the response of the model is shown when the flowtime is increased from 30 to 40 days. Shortages and expediting are virtually eliminated. The inventory levels reverse directions. By starting subassembly order processing earlier (40 days vice 30), component inventory levels are reduced, but with the extra planning flowtime, the subassembly inventory level increases. By increasing the planning estimate, flowtime buffer is created to absorb process variability.[4]

Why not just increase all planning flowtimes ? Whether buffer comes in the form of larger order quantities or longer flowtimes the end result is the same, larger inventory holding costs. In addition, an insidious problem often develops with this approach. It is the old "the job expands to fill the time available" syndrome. Instead of continuously improving the process and

---

[4] This is a frequent theme of Eliyahu Goldratt's in The Haystack Syndrome.

Table 5.15 - Flowtime Reduction Performance Chart

Flowtime Reduction		**Policy Alternatives**			**% Change**	
Performance Metric	**Base Case**	**Trial 1 (20)**	**Trial 2 (40)**		**Trial 1 (20)**	**Trial 2 (40)**
Average Order Quantity (units)						
Stringers	60	60	60		0%	0%
Doublers	83	83	83		0%	0%
Clips	83	83	83		0%	0%
Webs	83	83	83		0%	0%
Subassemblies	15	15	15		0%	0%
Average Inventory Level (units)						
Stringers	33	38	32		15%	-3%
Doublers	50	54	45		8%	-10%
Clips	54	58	46		7%	-15%
Webs	52	55	47		6%	-10%
Subassemblies	9	7	12		-22%	33%
Cycle Time (days)						
Stringers	36	37	36		3%	-3%
Doublers	40	44	38		10%	-5%
Clips	169	173	170		2%	1%
Webs	179	179	176		0%	-2%
Subassemblies	23	20	24		-13%	4%
Number of Expedite Actions (#)						
Stringers	2	3	0		50%	-100%
Doublers	0	1	0		INC	NC
Clips	1	1	0		0%	-100%
Webs	1	1	0		0%	-100%
Subassemblies	14	20	5		43%	-64%
Total	18	26	5			
Component Stockouts (#)						
Stringers	2	0	0		-100%	-100%
Doublers	0	0	0		INC	INC
Clips	1	0	0		-100%	-100%
Webs	1	1	1		0%	0%
Total	4	1	1			
Assembly Stockouts						
Total Number	5	12	1		140%	-80%
Cumulative Days	14	58	1		314%	-93%

reducing the flowtime required, the long, and inefficient process flows are institutionalized.

This "drifting goals" phenomena is diagrammed in Fig. 5.5. In this diagram, the goal is Boeing's desire to reduce its process flow allocations (i.e. - to move from a 30 day flowtime to 20 days using the simulation example). Currently, the actual condition of Boeing's processes will only yield a 30 day flowtime. The gap between the actual level and where Boeing wants to be is large.

This gap causes one of two possible loops to be followed. Boeing can start to improve its processes, or it can lower its expectations about flowtime improvement. Improving the process takes time; there are typically no overnight results. Once the process is improved, however; the process cycle times become more in line with the original goal, and so the gap closes.

Reducing the goal has an immediate effect of reducing the stress caused by attempting to meet the goal. This stress can come from the difficulty of taking a hard look at process improvement as well as from conditions such as product shortages which may increase as the process is altered. By saying, "we really don't need to get to 20 days; let's try 29.5 days to start out with", the goal is effectively eliminated and the gap closes very quickly. Unfortunately, the many pressures experienced within the Boeing production environment make this option the more likely.

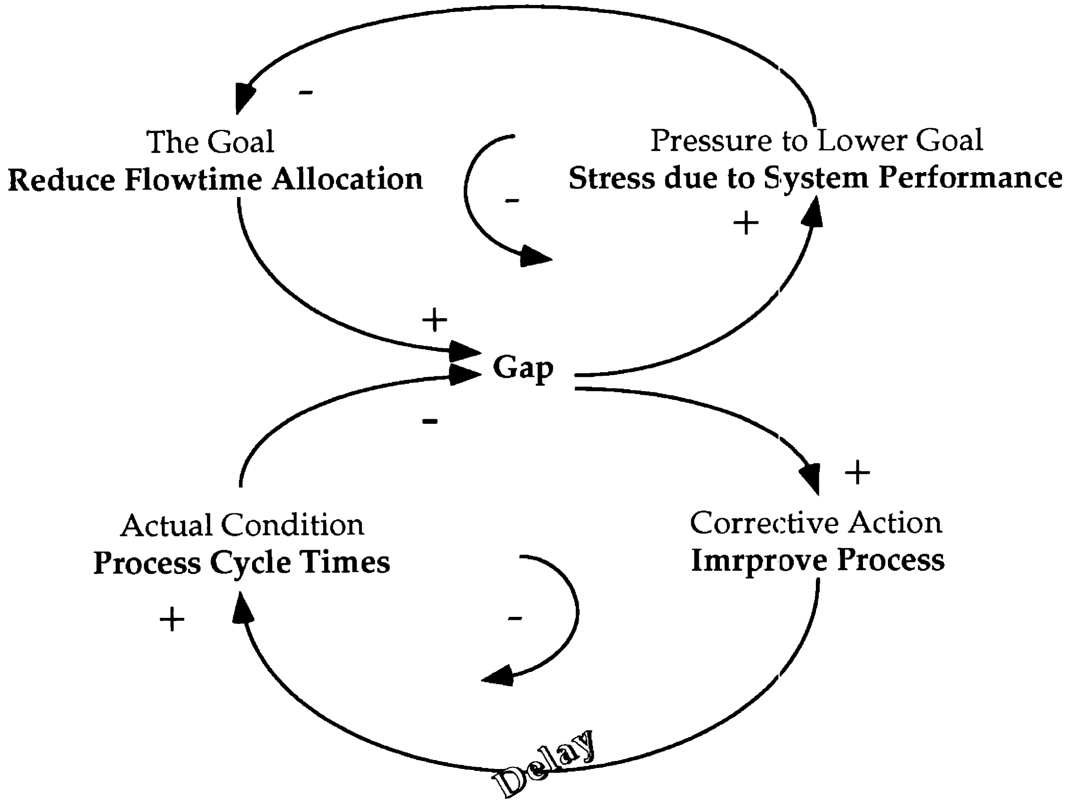


Figure 5.5 - Drifting Goals Archetype

Process Improvement

Taking a more positive approach and assuming that a permanent program dedicated to process redesign and continuous process improvement is implemented, how would this change system performance ? The process improvement assumptions used for model testing can be seen in Table 5.16.

Table 5.16 - Process Improvement Assumptions

Trial #1	Trial #2	Trial #3
30 %	40 %	50 %

The results of these tests can be seen in Table 5.17. In these tests, the process means for all manufacturing processes were reduced by the percentage indicated, but the variance in the process remained unchanged. This level of improvement could be the result of improvements in material flow, reduced setup times, and a variety of other techniques to reduce cycle time.

As a result of these modifications, up to a 93% reduction in the number of subassembly expedites and a 100% reduction in the number of subassembly shortages is realized. These changes create a built-in safety stock of subassembly inventory as shown in Figure 5.6. By reducing process cycle times, the ability to reduce inventory levels without fear of chronic shortages becomes a reality.

An interesting side effect of an improvement in the manufacturing process is that the average inventory level for all components increases. The reason this occurs is planning flowtimes have not been altered. The ordering system has not been updated to reflect the new process capabilities of the different work areas. Inventory ends up residing in storage longer than it did when the process cycle time was longer.

Based on the lack of process capability feedback to the ordering system, this is potentially a very real problem for Boeing management. If Boeing CQI efforts lead to improved processes but planning flowtimes are not adjusted, Boeing internal metrics on inventory would indicate that inventory holding costs are increasing. Boeing management might get the (wrong) impression that CQI efforts are not working.

Table 5.17 - Process Improvement Performance Chart

Process Improvement		**Policy Alternatives**			**% Change**		
Performance Metric	**Base Case**	**Trial 1 (30%)**	**Trial 2 (40%)**	**Trial 3 (50%)**	**Trial 1 (30%)**	**Trial 2 (40%)**	**Trial 3 (50%)**
Average Order Quantity (units)							
Stringers	60	60	60	60	0%	0%	0%
Doublers	83	83	83	83	0%	0%	0%
Clips	83	83	83	83	0%	0%	0%
Webs	83	83	83	83	0%	0%	0%
Subassemblies	15	14	14	14	-7%	-7%	-7%
Average Inventory Level (units)							
Stringers	33	36	37	40	9%	12%	21%
Doublers	50	52	55	57	4%	10%	14%
Clips	54	64	75	82	19%	39%	52%
Webs	52	60	70	73	15%	35%	40%
Subassemblies	9	10	10	13	11%	11%	44%
Cycle Time (days)							
Stringers	36	25	22	19	-31%	-39%	-47%
Doublers	40	30	28	23	-25%	-30%	-43%
Clips	169	120	102	81	-29%	-40%	-52%
Webs	179	128	111	92	-28%	-38%	-49%
Subassemblies	23	12	11	8	-48%	-52%	-65%
Number of Expedite Actions (#)							
Stringers	2	1	1	0	-50%	-50%	-100%
Doublers	0	0	0	0	NC	NC	NC
Clips	1	0	0	0	-100%	-100%	-100%
Webs	1	0	0	0	-100%	-100%	-100%
Subassemblies	14	11	9	1	-21%	-36%	-93%
Total	18	12	10	1			
Component Stockouts (#)							
Stringers	2	1	1	0	-50%	-50%	-100%
Doublers	0	0	0	0	NC	NC	NC
Clips	1	0	0	0	-100%	-100%	-100%
Webs	1	0	0	0	-100%	-100%	-100%
Total	4	1	1	0			
Assembly Stockouts							
Total Number	5	2	0	0	-60%	-100%	-100%
Cumulative Days	14	2	0	0	-86%	-100%	-100%

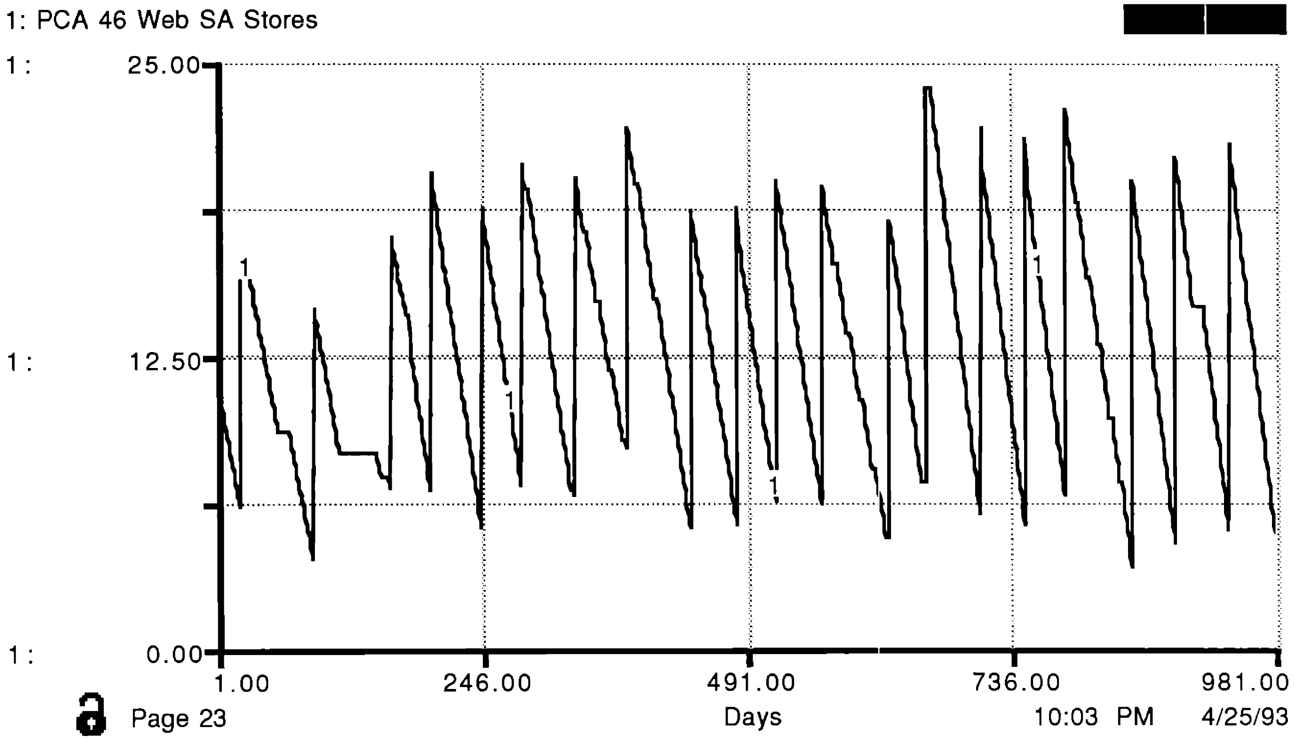


Figure 5.6 - Subassembly Stores - 50% Process Improvement

Timing of Deliveries

Currently detail and assembly ordering are not synchronized. The quantities vary depending on the time span chosen or the output from the EOQ algorithm. The schedules of the details and the next level assembly are tied only indirectly to one another,[5] but the implicit assumption within the Puget Sound Flow standards is that the schedules are synchronized. In this system, shortages are possible even when there are no other perturbations present.

As the reader may recall from a previous discussion, the Puget Sound Flow standard allocates the flowtime seen in Fig. 5.7 for this assembly. What this means is that 30 days prior to running out of assemblies in Control Code 352, Renton Lot Time needs to start building its next lot. Stringer production needs to start 71 days (50 days + 21 day part fabrication) in advance of running

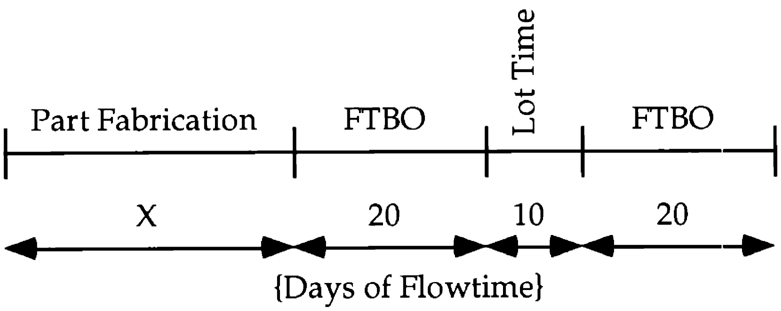


Figure 5.7 - Puget Sound Flowtime

---

[5] All details and assemblies are tied to the schedule of the control code that the final assembly is used in. For the assembly used in this project, the schedule in Control Code 352 drives the detail and assembly schedules.

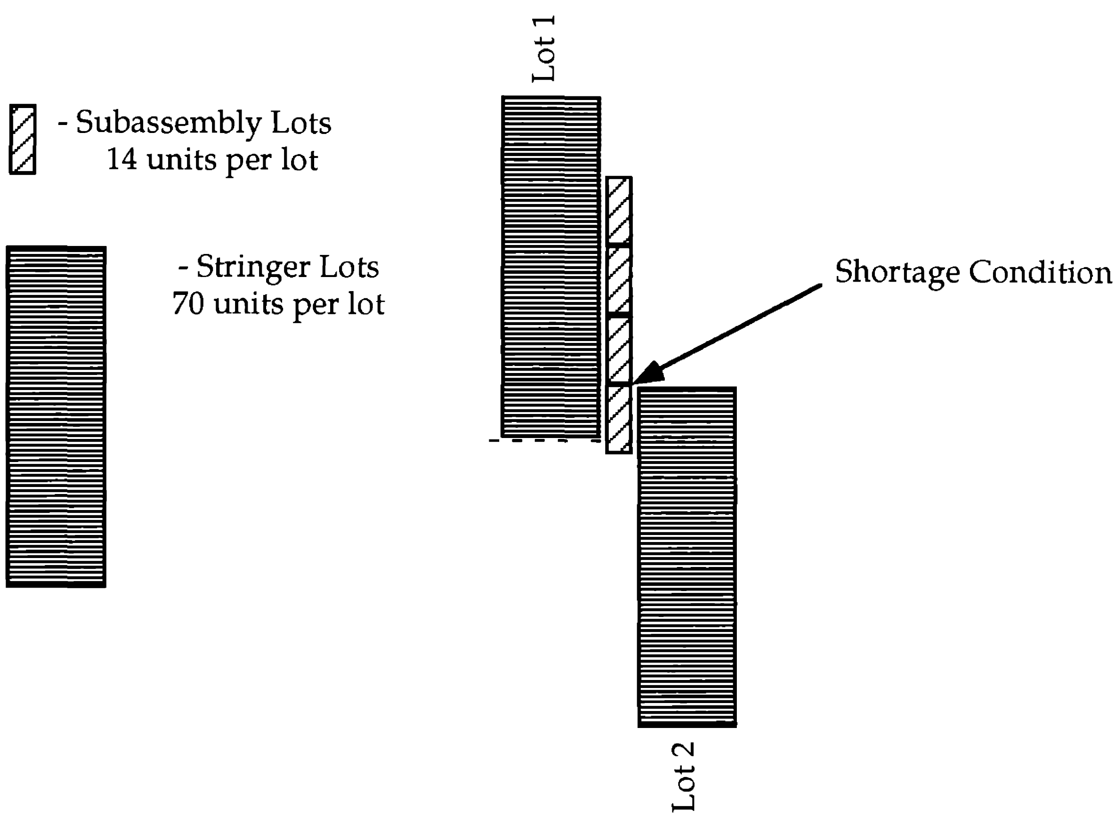


Figure 5.8 - Synchronization Problems within the Existing Ordering System

out of stringer inventory. Unfortunately, while the Puget Sound Flow standard assumes that both inventory levels are synchronized to the same point; they are not. A diagram that helps to explain this problem can be seen in Fig. 5.8. In this example, stringer lots are built in quantities of 70, and lot time assemblies are built using a 42 day time span which equates to roughly 14 units[6]. What has happened here is that Lot Time is attempting to build its next lot of 14 assemblies, but there are not sufficient stringers in the bins to build a lot of 14. The next lot of stringers will not be in for another 12 days.

---

[6] A 3 day cycle in Control Code 352 is assumed.

The ordering system will make the outstanding stringer order SOS, and Lot Time will either make what they have available or wait for the newly designated "hot" stringer order to arrive.

Synchronizing Ordering

One possible approach to correct this problem, and a few others, would be to create a tighter linkage between assembly and detail ordering. By making the detail order quantity a multiple of the assembly quantity and generating detail order signals based on detail inventory levels in assembly areas, a synchronized ordering process might be possible.

One method of implementing this approach for the assembly used in this analysis would be with a three bin system like the one described below. Using

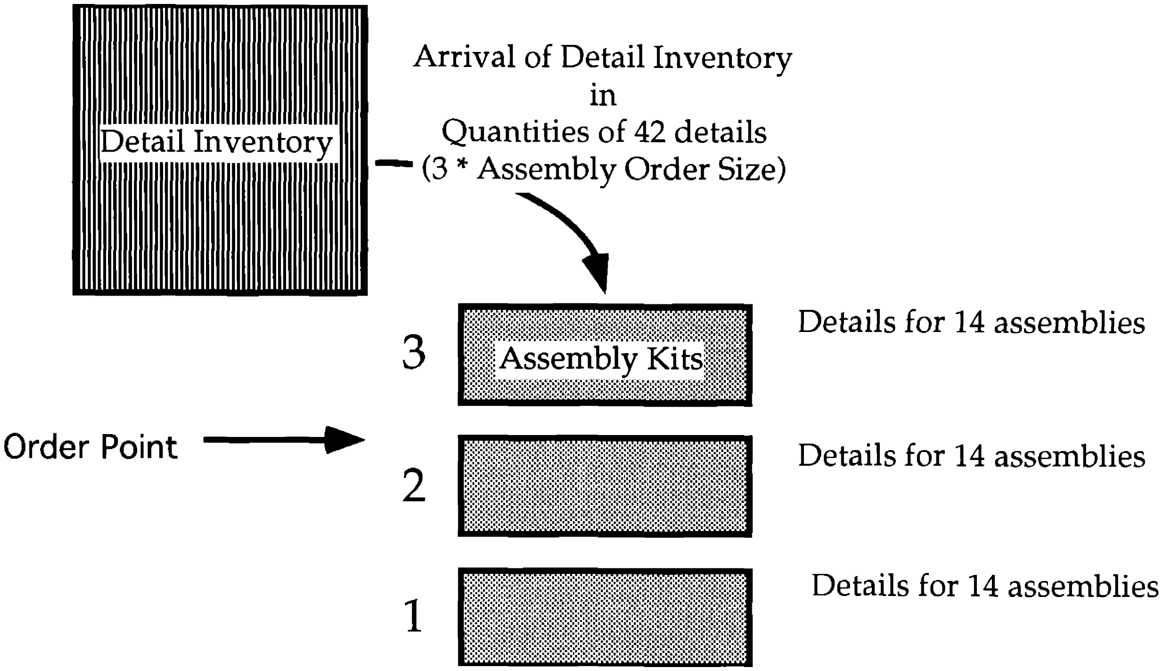


Figure 5.9 - 3 Bin System

the assembly from this project as the example, the time span used for assembly ordering is 42 days (14 units). Under this new ordering policy, the detail order quantity would be three times this amount (a 3 bin quantity) or 42 units. In Figure 9, the detail inventory arrives and is immediately used to fill three kits (1-3). Kit 1 will be consumed first by the Renton Lot Time area. As soon as the second kit is removed from the rack, orders go out to the detail producers (build 42 more details). While the new detail lot is being produced, Kit 2 is consumed and Kit 3 is removed for assembly. Based on the detail process times, some detail orders may arrive before Kit 3 is removed, others after.

A 3 bin approach was implemented in the model by making a few minor structural modifications. The parameter values used for this test can be seen in Table 5.18.

Table 5.18 - 3 Bin System Parameter Values

Component	Order Time Span (days)	Order Point
Stringer	126	84
Doubler	126	84
Clip	126	84
Web	126	84

The response of the model to these changes can be seen in Table 5.19. The average inventory level decreases by 9 to 42% for the detail components. The number of expedite actions drops for detail components and remains constant for subassemblies. The number and cumulative time of subassembly stockouts drops 40% and 36% respectively. The lack of expediting and prioritization of work orders causes the cycle time of stringers and doublers to increase relative to the base case.

Based on these results, it appears that there might be additional buffer available in stringer and doubler ordering. The second trial shifts stringer and doubler ordering to a 2 bin system. The parameter values used for this trial can be seen in Table 5.20.

Table 5.20 - 2/3 Bin System Parameter Values

Component	Order Time Span (days)	Order Point
Stringer	84	42
Doubler	84	42
Clip	126	84
Web	126	84

Supplier detail production was left on a three bin system; doubler and stringer production were changed to a two bin system. The model was modified to incorporate these changes, and the response of the model can be seen in Table 5.19. A further reduction in inventory levels is realized (37 - 70%), but a price is paid. Stringer and doubler expedite action increases, and doubler shortages increase. Subassembly stockout performance is unchanged from the 3 bin case. By synchronizing assembly and detail ordering, inventory levels and subassembly stockouts can be reduced even when inventory and quality variability is present.

While this approach appears feasible, several things must happen. Supplier ROLTs must be reduced. The effect of more frequent orders on the process distribution of internal and external suppliers must be addressed. A whole host of other implementation issues would also have to be reviewed. Once again the point of this policy analysis was not to define the optimum number of bins to use, but rather to determine if this approach is feasible and if it would have any potential benefit for Boeing. Based on the model results, it looks feasible (with the caveats mentioned previously), and even if the bin count was set so high that there was no real inventory reduction, the bin system would be preferable to the current system because it would synchronize detail and assembly production leading to fewer stockouts.

Table 5.19 - Bin System Performance Chart

Flowtime Reduction		**Policy Alternatives**			**% Change**	
Performance Metric	**Base Case**	**3 bin**	**2/3 bin**		**3 bin**	**2/3 bin**
Average Order Quantity (units)						
Stringers	60	43	29		-28%	-52%
Doublers	83	43	28		-48%	-66%
Clips	83	43	43		-48%	-48%
Webs	83	43	43		-48%	-48%
Subassemblies	15	14	14		-7%	-7%
Average Inventory Level (units)						
Stringers	33	30	15		-9%	-55%
Doublers	50	35	15		-30%	-70%
Clips	54	34	34		-37%	-37%
Webs	52	30	30		-42%	-42%
Subassemblies	9	9	9		0%	0%
Cycle Time (days)						
Stringers	36	39	34		8%	-6%
Doublers	40	43	39		8%	-3%
Clips	169	48	48		-72%	-72%
Webs	179	55	55		-69%	-69%
Subassemblies	23	20	20		-13%	-13%
Number of Expedite Actions (#)						
Stringers	2	0	7		-100%	250%
Doublers	0	0	10		NC	INC
Clips	1	0	0		-100%	-100%
Webs	1	1	1		0%	0%
Subassemblies	14	14	14		0%	0%
Total	18	15	32			
Component Stockouts (#)						
Stringers	2	0	1		-100%	-50%
Doublers	0	0	3		NC	INC
Clips	1	0	0		-100%	-100%
Webs	1	1	1		0%	0%
Total	4	1	5			
Assembly Stockouts						
Total Number	5	3	3		-40%	-40%
Cumulative Days	14	9	9		-36%	-36%

Blue Sky

In this final test scenario for the model, significant improvements are assumed for Boeing's manufacturing process within the bin system and order point framework. The parameter values for this test are listed in Table 5.21.

Table 5.21 - Blue Sky Parameter Values

	Blue Sky	Base Case
System Variability	0% Inventory Losses/Additions	$\mu = .03; s = .02$
	0% Quality Losses	$\mu = 0; s = .02$
Subassembly	Mean = 1 day	Mean = 30
	Std Dev = .25 day	Std Dev = 8.2 days
	Time Span = 21 days	Time Span = 42 days
Stringers & Doublers	Mean = 1 week	Mean = 39-45 days
	Std Dev = 1 day	Std Dev = 2 days
	Time Span = 44 days	Time Span = 252 days
Clips & Webs	Mean = 2 weeks	Mean = 185 days
	Std Dev = 2 days	Std Dev = 3-9 days
	Time Span = 84 days	Time Span = 252

While these parameter values represent a dramatic improvement for Boeing, they are realistic and are really quite mediocre compared to the capability of most lean producers. [Merli, 1990] [Schonberger, 1987] [Suzaki,1987] The performance improvement that results from these changes can be seen in Table 5.22 and Figures 5.10-5.14. Inventory reductions of 33 to 68% are realized over the base case, and virtually all expedite and stockout activity is eliminated.

The reduction in average inventory level demonstrated in this simulation run increases the annual inventory turns from ~1.5 to ~3.5 for externally supplied components and from ~1.7 to ~5.25 for internally supplied components. While many lean producers have inventory turns in excess of 20 times per year, 3.5 and 5.25 turns per year still represents a significant improvement for Boeing. In fact, Mr. Ron Woodard,[7] has stated that a .65 increase in the number of turns from the current level at Renton Division represents dollar savings equal to Renton Division's total profit for the last year.

Many things must happen before Boeing can realize the benefits reviewed in this policy test. Boeing must move toward product focused, small lot production. Enabling techniques, like setup time reduction, must be vigorously pursued to make small lot production possible. Product quality and inventory accuracy must approach 100% levels. The separate production islands need to be coupled to one another, and in some cases consolidated, to reduce the delays in feedback of information between areas. While Boeing has many improvement efforts underway, a great deal still must be accomplished before Boeing can match the Blue Sky parameter values on a global level. As Ron Woodard recognizes, it is certainly worth the effort.

---

[7] Mr. Ron Woodard, General Manager of Renton Division, in a speech before the Door Product Center on 10/12/92.

Table 5.22 - Blue Sky Performance Chart

Blue Sky		Policy Alternatives		% Change
Performance Metric	**Base Case**	**Trial 1**		**Trial 1**
Average Order Quantity (units)				
Stringers	60	16		-73%
Doublers	83	16		-81%
Clips	83	29		-65%
Webs	83	29		-65%
Subassemblies	15	8		-47%
Average Inventory Level (units)				
Stringers	33	16		-52%
Doublers	50	16		-68%
Clips	54	24		-56%
Webs	52	24		-54%
Subassemblies	9	6		-33%
Cycle Time (days)				
Stringers	36	5		-86%
Doublers	40	5		-88%
Clips	169	9		-95%
Webs	179	9		-95%
Subassemblies	23	1		-96%
Number of Expedite Actions (#)				
Stringers	2	0		-100%
Doublers	0	0		NC
Clips	1	0		-100%
Webs	1	0		-100%
Subassemblies	14	1		-93%
Total	18	1		
Component Stockouts (#)				
Stringers	2	0		-100%
Doublers	0	0		NC
Clips	1	0		-100%
Webs	1	0		-100%
Total	4	0		
Assembly Stockouts				
Total Number	5	0		-100%
Cumulative Days	14	0		-100%

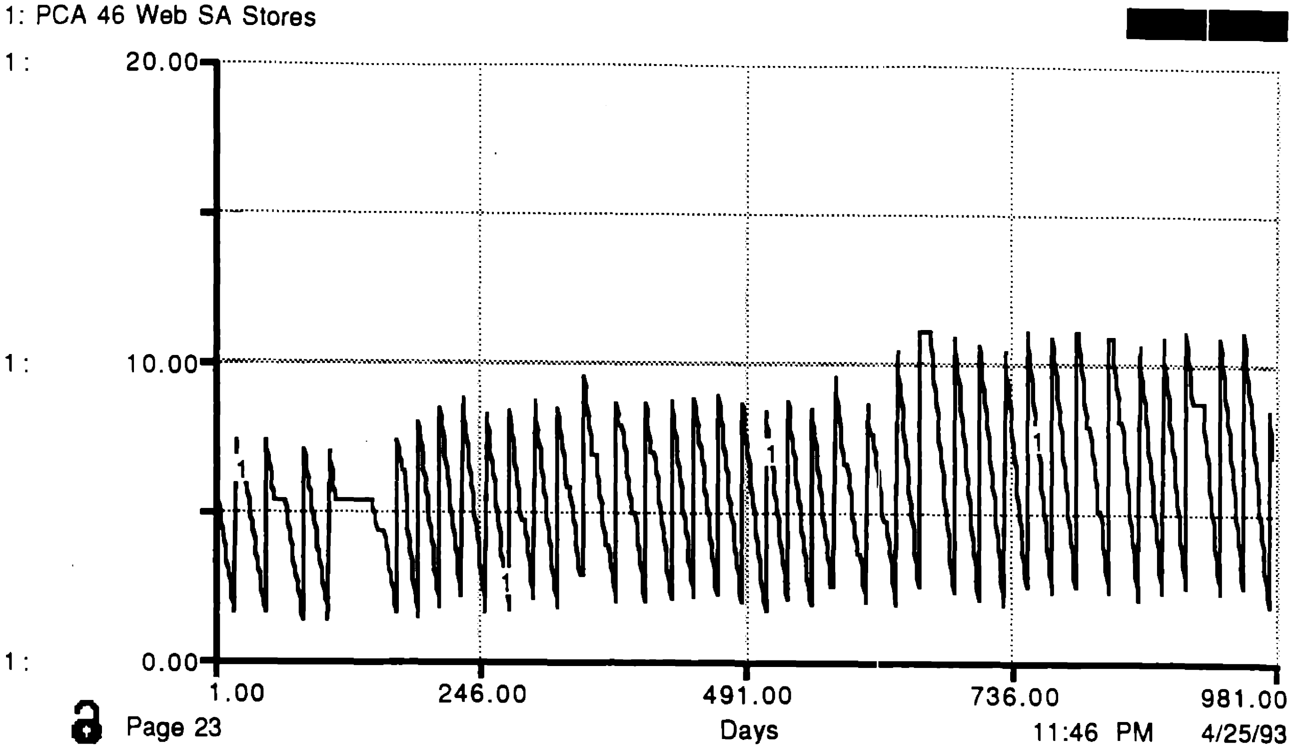


Figure 5.10 - Subassembly Stores - Blue Sky

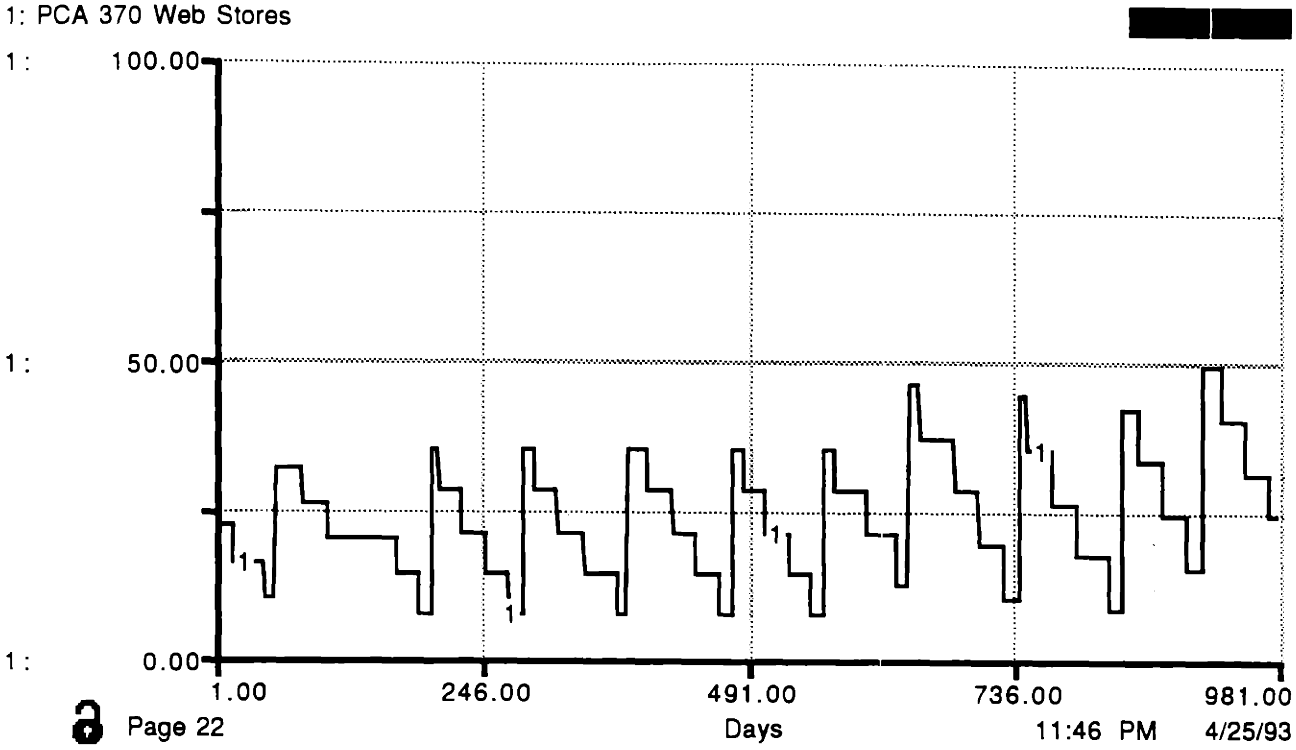


Figure 5.11 - Web Stores - Blue Sky

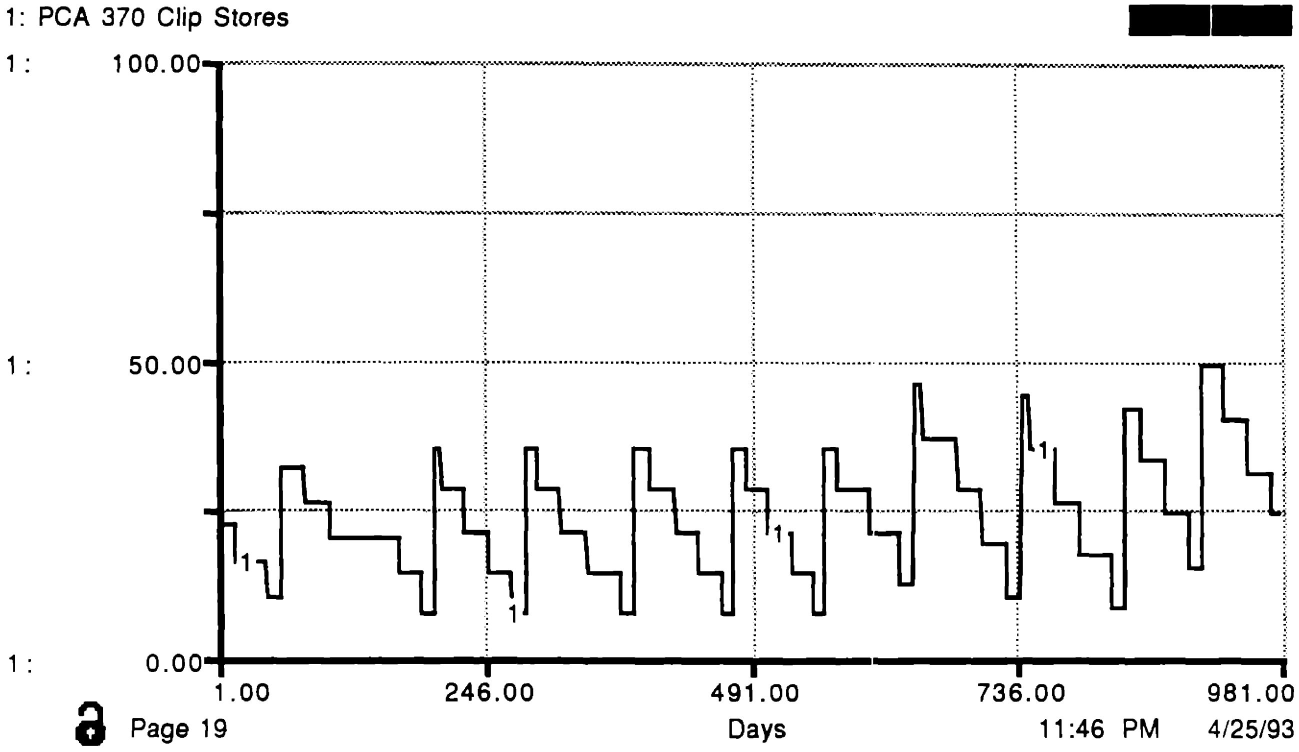


Figure 5.12 - Clip Stores - Blue Sky

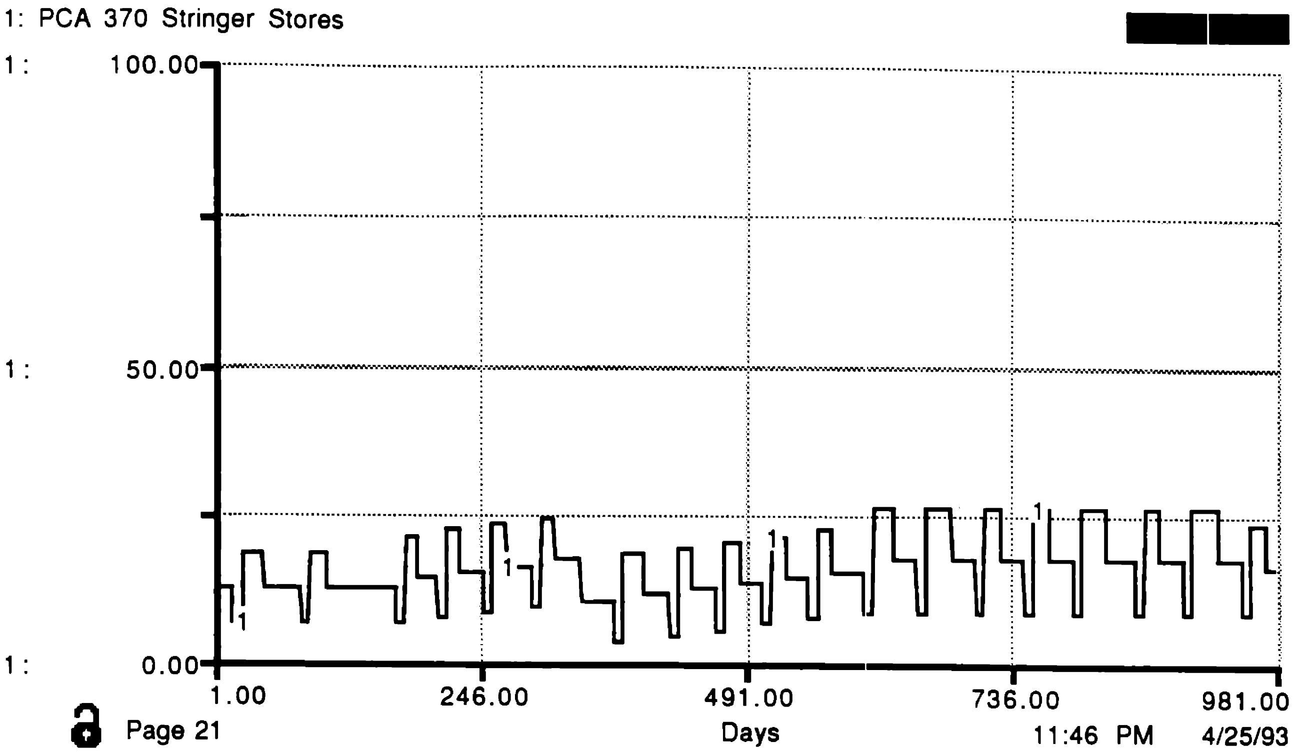


Figure 5.13 - Stringer Stores - Blue Sky

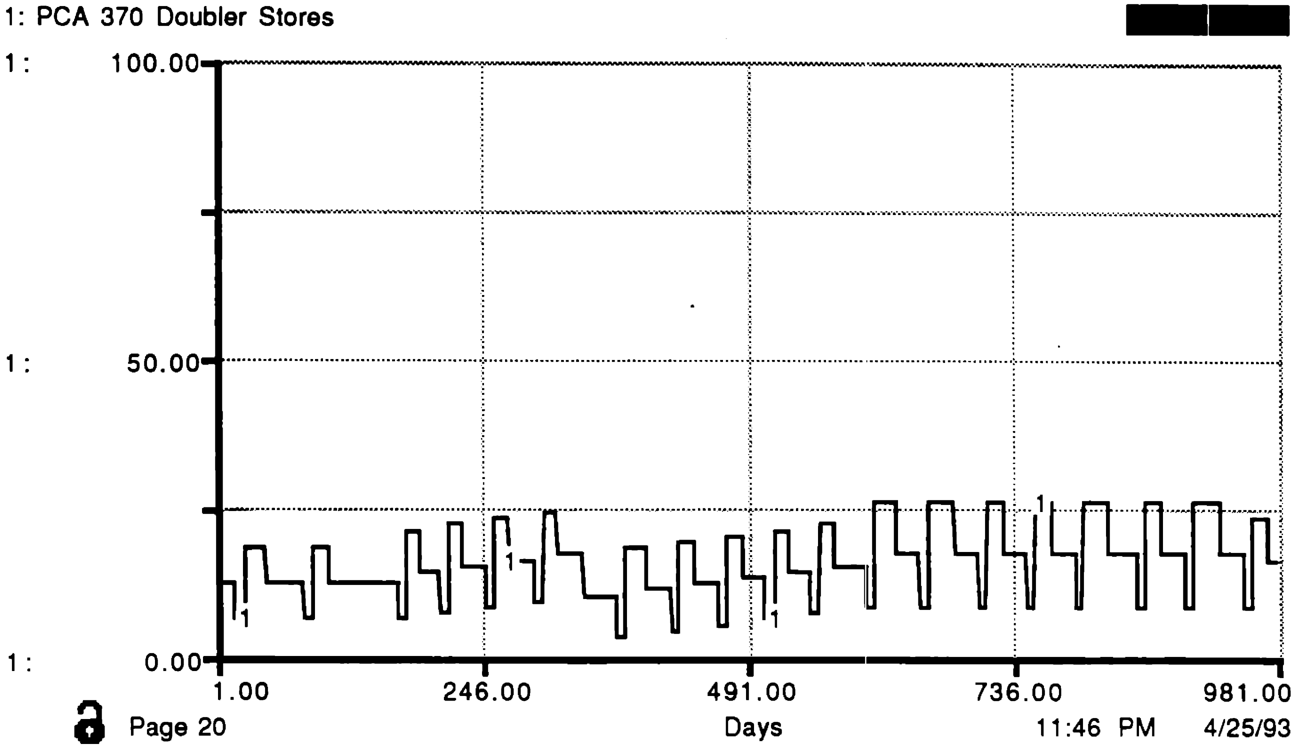


Figure 5.14 - Doubler Stores - Blue Sky

Conclusions

The model allows for experimentation with different ordering policies prior to any attempt at implementation. The response of the model to the different policy alternatives should be viewed more as a compass direction than a GPS[8] navigational fix. While the model should indicate whether a policy has desirable or undesirable consequences, precise prediction of the actual system's response should not be expected. Using the compass headings from the previous analysis, a summary of policy results is presented in Table 5.23.

Table 5.23 - Summary of Policy Analysis

Policy	Benefits	Problems
Reducing Order Size (independent of constraints)	• up to 58% reduction in inventory levels with stable response	• does not improve the underlying process • large cuts in order qty lead to increases in expedites and stockouts
Reducing Order Size (within constraints)	• inventory reduction of 42-64% without significant increase in expedites • small lot sizes increase production flexibility & reduce stockouts	• does not improve the underlying process • cuts to 42 day Time Span cause increase in expedites
Reducing ROLTs	• 54-65% inventory reduction with stable response	• requires negotiation of 60 day ROLTs with suppliers • pushing suppliers beyond their process capability leads to long stockouts
Inventory Accuracy	• demonstrates the need for accurate inventory counts	• delays in discovery & size of error lead to longer & longer stockouts

---

[8] This reference is to the Global Positioning System, a constellation of satellites used for precise position determination.

Product Quality	• demonstrates the cascading impact of poor product quality on operations	• with 30% quality loss there are 14 stockouts lasting 108 days • leads to "green lines" and other production disruptions
Planning Flowtime	• increasing flowtime (40 days vice 30) creates greater stability leading to fewer expedites and stockouts	• reducing flowtime beyond current process capabilities leads to instability • increasing flowtime leads to "institutionalization" of longer cycle times and higher inventory carrying costs
Process Improvement	• 50% reduction in process cycle time leads to 93% reduction in expedite action and eliminates stockouts • provides opportunity to safely reduce inventory levels and planning flowtimes	• requires significant improvement in production processes • leads to increase in average inventory levels until planning flowtimes are reduced
Bin System	• 3 bin system leads to 9-42% inventory reduction with 17% fewer expedites and 40% reduction in stockouts • 2/3 bin system leads to 37-70% inventory reduction with 40% fewer stockouts	• 2/3 bin system causes a 78% increase in expediting • requires 60 day supplier ROLTs • requires rearrangement of PCA area to accommodate bin system implementation
Blue Sky	• leads to 33-68% inventory reduction with 1 expedite and no stockouts • equivalent to moving from 1.5 turns to 3.5 for supplier parts and 1.7 turns to 5.25 for internally produced parts	• requires a system's approach to integrate Boeing's production islands • requires significant improvement in cycle times and quality for all suppliers (internal & external)

Based on the trial runs conducted, it is clear that the process capabilities that exist within the manufacturing environment govern the type of ordering policies that can reasonably be attempted. Reducing ordering quantities or flowtime allocations without considering the capability of the underlying manufacturing processes is a plan that is potentially fraught with peril. Reduction of inventory levels and planning flowtimes should be done in concert with process improvement activities.

In Figure 5.15, Boeing's current process capability is represented. The process mean takes the form of a normally distributed bell curve which has a large spread due to the variability in the process. This variability results from problems such as machine breakdown, labor availability, inventory accuracy, product rejections and rework due to poor quality, engineering changes, and unanticipated requirements to name a few. The resulting instability that can develop from high variability was demonstrated during model simulations with increasing levels of inventory loss and quality rejections as shown in Figures 5.17 & 5.18. For the purposes of this discussion, ordering system instability is characterized by frequent expedite action, product shortages, or excess inventory.

In Figure 5.17, inventory loss from the storage bins leads to an exponential increase in shortages and an increase and leveling out of expedite actions. Quality losses, shown in Figure 5.18, leads to an exponential increase in shortages and a fairly linear increase in expedite actions. The points which mark a more significant degradation in system performance appear to occur at the 7% level for inventory loss and the 10% level for quality loss. While 0% inventory or quality loss is the goal, it is important to avoid regions where system performance will experience significant variation.

The box that encompasses a +/- 3σ variation from the process mean in Figure 5.15 & 5.16 represents a stable region of operation for the ordering system. If the process mean increases and the planning flowtime does not account for this shift, expedite action and product shortages will result. This dynamic was seen in model simulations and is captured in Figure 5.19. As planning flowtime is reduced below 25 days, the system response becomes less stable. Coversely, as the flowtime was increased past 35 days, expedite action and shortages were eliminated. Increasing the planning flowtime, however, can substitute one form of instability for another if expedites are reduced but inventory levels increase. Currently, the eight month ROLTs with most suppliers (see Fig. 2.5) are representative of this type of instability. What is desired is to accurately match the planning flowtime with the actual process capability while continually shifting the process mean toward zero and eliminating all sources of process variability.

This goal is represented by Figure 5.16. The distribution shrinks as variability is removed and moves left on the time line indicating a reduction in the mean cycle time. The planning flowtime shrinks accordingly. Ideally, the process cycle time would become very close to zero and would become a pulse representing zero variability. With manufacturing processes this fine tuned, concepts such as "one-piece flow" can be turned into reality.

Real movement toward this goal requires a systematic approach to process improvement across all elements of production. Currently, Boeing lacks the ability to look at process improvement from a global perspective. It is up to each of the "production islands" to improve their own processes. All efforts are local and will undoubtedly yield local optimums.

One of the main reasons that a global approach needs to be used is because of the misleading results that can be experienced. Counterintuitive results, as seen in the model simulation trials, can cause managers to abandon process improvement activities. For example, if the process mean shifts to the left but the planning flowtime is not adjusted accordingly, excess inventory will be the result. Additionally, it is also possible that one poorly performing detail production process can cause the inventory levels of other detail components to rise even when their individual processes are performing adequately. If a manager is unaware of system wide impacts of local activities, he or she is likely to be confused by these type of results and eliminate activities that are beneficial. The gains for Boeing are potentially too significant to let this happen.

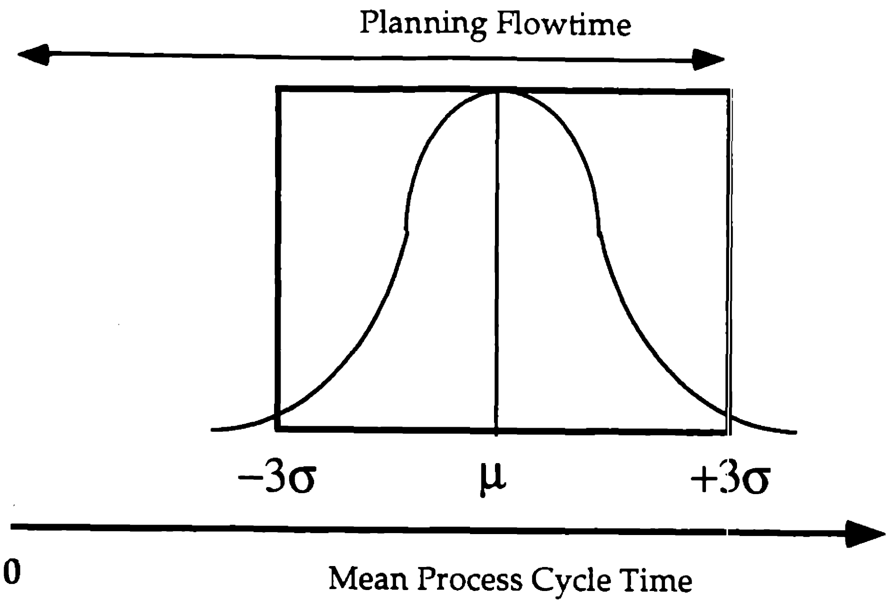


Figure 5.15 - Current Process Capability

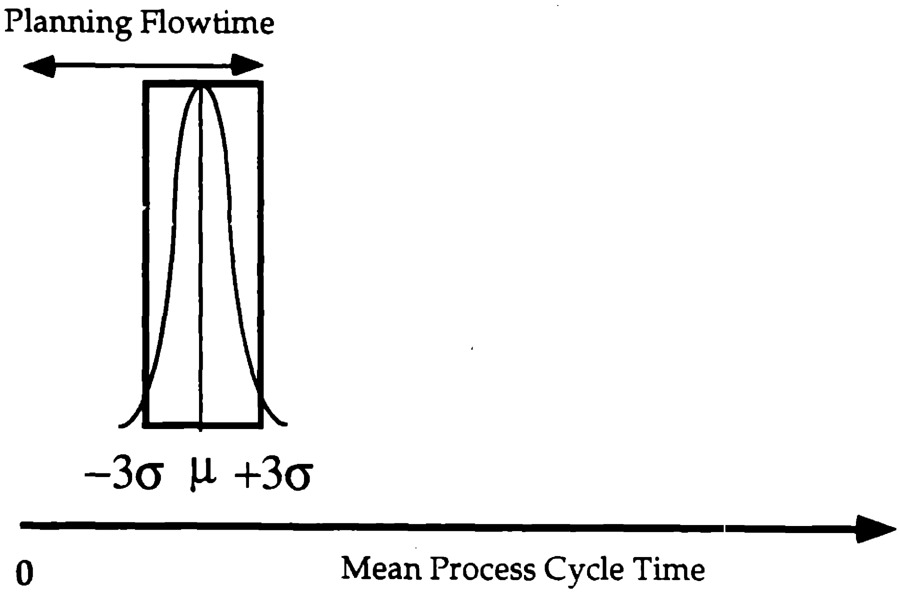


Figure 5.16 - Desired Process Capability

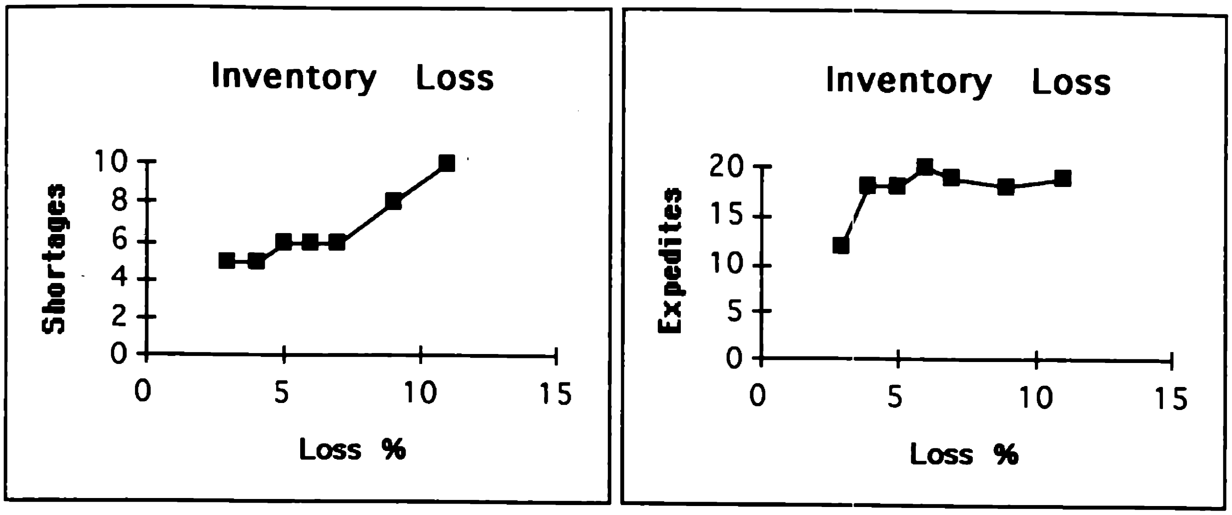


Figure 5.17 - Impact of Inventory Loss

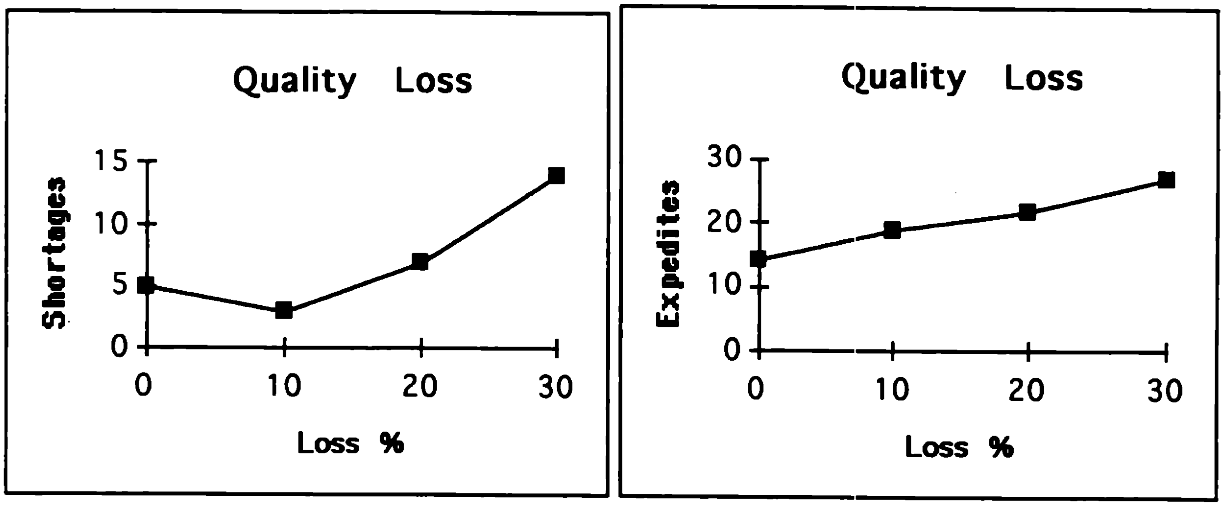


Figure 5.18 - Impact of Quality Problems

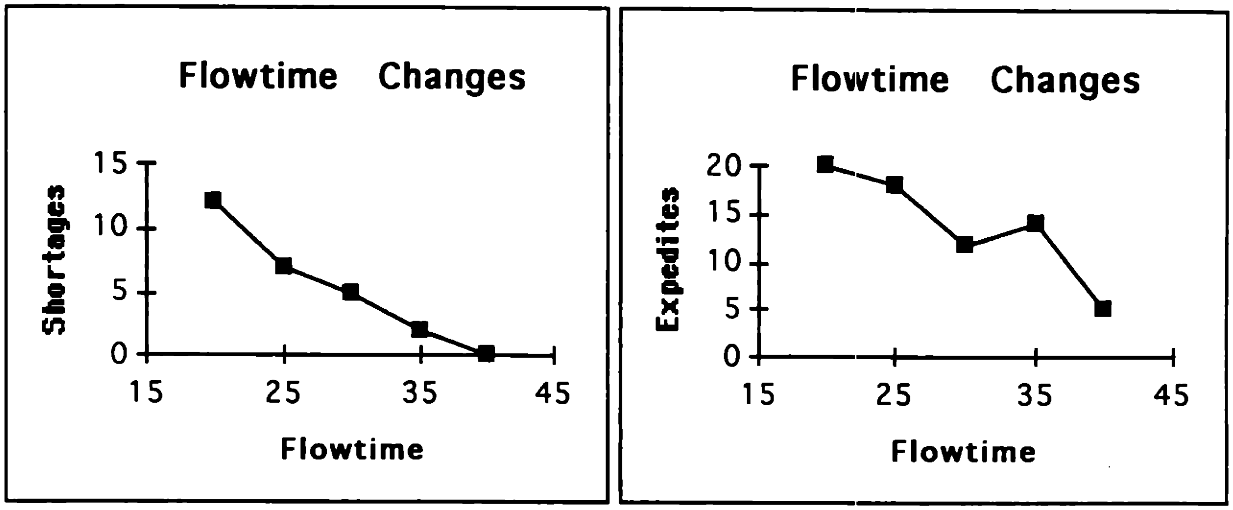


Figure 5.19 - Impact of Changes in Planning Flowtime

Chapter 6 - Conclusions and Recommendations

Conclusions

As a summary of the first five chapters, I list and discuss the major conclusions of my research. For a more detailed review of the specific subject, please refer to the chapter indicated in parentheses.

Boeing needs a New Approach to Aircraft Manufacture (Chapter 1)

While Boeing remains the dominant airframe manufacturer, its high production costs make it vulnerable to market share loss as airlines seek less expensive alternatives or better financing arrangements that Boeing is unable to provide. Currently, Boeing's aircraft production strategy creates excessive "linkage costs" and non-value added activity. This approach has limited room for improvement compared with the lean manufacturing approach that has been adopted in other industries.

The Boeing Ordering System is Overly Complex (Chapter 2)

The ordering system that Boeing uses is really a collection of systems. Over 50 different hardware platforms and software applications are required to implement ordering actions. The different age and capability of these systems requires an expensive maintenance infrastructure to keep them operational and to implement continual performance modifications. To operate the different systems a complicated vocabulary and unique process steps must be mastered for each system. This complexity makes human error a costly problem leading to wasteful duplication of administrative ordering tasks. Another result of system complexity is that very few people within Boeing understand how all elements of the ordering system operate making intelligent change a difficult task.

Current Ordering System Elements Institutionalize Bad Habits (Chapter 2)

Embedded within the ordering system are policies that lead to the development of undesirable behavior. Puget Sound Flow Standards, which specify the long planning flowtimes, reinforce a belief that it really takes several months to process an aircraft part when the actual touch time is less than one day. The accumulation of these prior experiences make resistance to lean production techniques much harder to overcome. Protective ordering, that is used to limit exposure to product losses, has the side effect of hiding quality problems. With protective ordering, appropriate problem solving techniques will not be used, and product quality problems will continue. Economic Order Quantity (EOQ) ordering, that is used for most detail components, generates large lots, making setup time reduction unnecessary and institutionalizing large inventory holding costs. Eight month reorder lead times (ROLTs) for most suppliers also increase inventory holding costs and make Boeing more vulnerable to engineering changes. All of these actions by the ordering system cover up problems rather than exposing them so that they can be eliminated. Because many of these functions are embedded within the ordering system, production area management is put into a reactive mode with respect to their own production requirements.

Root Causes of Product Shortages (Chapter 2)

The following problems were determined to be the primary root causes of product shortages:

- ordering changes & errors
- product quality problems

- process flowtime variation
- inventory accuracy
- product configuration variation.

Root Causes of Excess Inventory (Chapter 2)

The following problems were determined to be the primary root causes of excess inventory:

- ordering errors
- engineering changes
- ordering policy
- scheduling policy
- inventory accuracy
- manufacturing policy

Benefits of System Dynamics Analysis (Chapter 3)

The System Dynamics method has two primary benefits: 1) it allows for a much deeper understanding of the system, and 2) it provides a managerial "microworld" where policies can be attempted prior to actual implementation.

Problem of Data Overload and Information Scarcity (Chapter 3)

A continual problem through out the research was the availability of useful information. The business systems are full of raw data, much of it of questionable accuracy, but useful and simple metrics to chart system performance were not readily available and to my knowledge have not been developed. If there are useful and accurate metrics, they are hidden within a plethora of minutia.

Model Validation (Chapter 4 and Appendix A-4)

The System Dynamics model developed for this project was validated in several ways. The structural representation was reviewed, in preliminary form, by several experts on the Boeing ordering system. The behavior of the model was analyzed against historical data, and this analysis yielded the following results:

- the absolute error was between 0-3%
- results of t-test analysis failed to reject the null hypothesis that the model data matched the historical data
- the Mean Absolute Percent Error (MAPE) indicated that doubler structure and/or parameter values need improvement.

Based on these results and others, I concluded that the model accurately reproduces the reference behavior of the actual system. The most important point, however, is that system dynamic's models are never completely validated; they continue to improve as knowledge of the systems that they represent increases.

Policy Analysis (Chapter 5)

Several different policy tests were conducted with the model. A brief summary of the major results is listed below. For a more detailed review, please refer to Table 5.23.

- up to a 65% reduction of inventory levels is possible from internal suppliers without a negative impact on system stability
- up to a 65% reduction of inventory levels is possible from external suppliers if ROLTs drop to 60 days
- inventory accuracy and product quality problems increase system instability
- delays in discovering inventory losses lead to longer stockouts
- product quality losses lead to "green line" work around with their disruptive effects
- planning flowtimes need to cover the $+3\sigma$ variation to prevent system instability
- improving process cycle times improves system stability and allows for reduction of inventory levels
- synchronized detail and assembly ordering leads to a 40% reduction in stockouts even as inventory levels are reduced

- inventory turn ratios on the order of 3.5 for supplier parts and 5.25 for internally produced parts are possible with process improvements
- counterintuitive results can confuse management when a good understanding of system dynamics does not exist

The potential for significant financial benefit exists for Boeing if it embraces a leaner production approach. The increase in inventory turn rates that are possible, while not that impressive for most lean manufacturers, would yield millions of dollars of annual savings for Boeing.

Recommendations

From my research on Boeing's Ordering System and insight gained from development and test of a system dynamics model of that system, I have several specific recommendations.

Simplify the Ordering Process

The following four recommendations are concerned with simplifying, what is an extremely complex, ordering process.

Use Time Span Ordering for all Components

Based on my analysis of the ordering policies used by Boeing, I believe that there are many negative aspects to the Economic Order Quantity algorithm (as discussed in Appendix B-1) used for most detail ordering. Using time span ordering for all components will have 3 primary benefits: 1) simplify the ordering process, 2) provide work centers with a more accurate forecast of future production requirements, and 3) provide an incentive to reduce inventory by challenging work centers to set new and lower time span goals. Initially the time span parameters chosen for detail components need not make substantial cuts in order quantity. For example, if the typical EOQ lot size for a component is 70 units, a time span that yields approximately 70 units could be chosen. The reason for doing this is to prevent production disruption during the transition from EOQ ordering to time span ordering for detail components. After the transition is complete, time span parameters can be reduced in conjunction with process improvement activities. Eventually detail time spans will be related to parent assembly time spans when detail and assembly ordering is synchronized.

Synchronize Detail and Assembly Ordering

As was clearly demonstrated during model policy analysis, detail ordering should be tied to assembly use rates. Detail order quantities should be a simple multiple of assembly order quantities. For example, if assemblies are built using a 42 day time span, details should be supplied in 84 or 126 day time span quantities. Visual systems, like bin storage, should be used to signal the release of the next detail order when the order point is reached. By ordering based on what is actually in the bin versus what is in the ordering system's electronic ledger, an entire sequence of non-value added steps and sources of error can be eliminated. A more visual control system will also have the benefit of shortening the delays in discovering component shortages and maintaining accurate inventory accuracy.

Use Simple Safety Stock Levels

Within the current ordering system there are several different parameters which lead to protective ordering. Effectively, these parameters create safety stock, but it is safety stock of an unknown quantity to the manager and workers in a production area. I recommend that Boeing move to a simpler and more visible form of safety stock. For example, with the bin system concept, the bottom kit would contain the safety stock as shown in Figure 6.1. When safety stock was used, it would be immediately replenished on the next order. The primary benefit of this concept is that it takes the nebulous and unknown safety stock levels of the current system and makes them very visible. Ideally, you want to get to the point where all safety stock can be eliminated, but the only way to do that is if you know what your current safety stock level is. This approach provides a visible incentive system. Every production worker will know that safety stock means excess inventory, and process improvement means that it will be possible to reduce safety stock levels and the excess inventory associated with it.

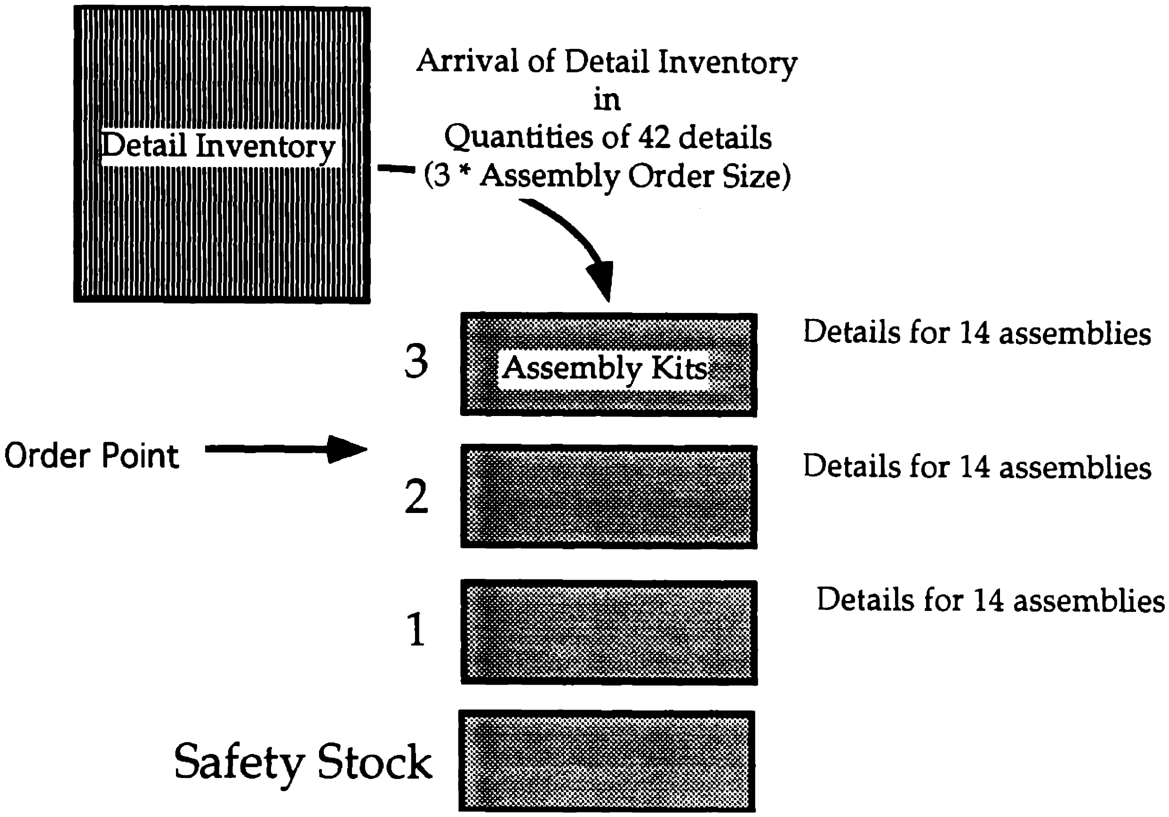


Figure 6.1 - 3 Bin System with Safety Stock

Define a New Business System Architecture

The current business systems used by Boeing for ordering and manufacture of commercial aircraft suffer from many problems: they are 1) not integrated, 2) overly complex to use, 3) based on outdated technology, and 4) are very expensive to maintain. Boeing needs to define a new business system architecture which will be flexible, open, and allow for localized control where desirable. The bin system provides one avenue for movement away from a heavy reliance on large computer systems for ordering action, but Boeing will obviously need to continue to store and transfer large amounts of data. A fresh look is required at what data and information is really needed to build an aircraft and what is superfluous.

Process Improvement

The following four recommendations are concerned with improving the capability of Boeing's processes. Ultimately, the key to inventory reduction, and the elimination of expedite actions and product shortages depends on improving the manufacturing process. No magical order quantity or sophisticated business system will solve production problems if the underlying manufacturing process is poorly designed.

Reduce Supplier Reorder Lead Times (ROLTs)

As demonstrated through model policy analysis, the delays in supplier response magnify the problems associated with inventory accuracy, product quality, and engineering changes. As the supplier base is rationalized, an opportunity exists for Boeing to work in partnership with its suppliers to significantly reduce ROLTs. The potential exists for a 65% reduction in inventory levels if a 60 day supplier ROLT could be achieved.

Application of Lean Production Techniques

Significant changes in how Boeing manufactures aircraft are required if it is to become a lean producer. Plant layouts need to be changed, and smaller and greater number of equipment need to be purchased to move toward a more product focused versus a more functionally focused company.

Many of the methods used by lean manufacturers have direct application within Boeing, and do not require expensive changes. For example, one example of point of use storage for tooling using the bin concept would be to hang the tooling on the bin. When the next assembly kit is to be delivered to the work area, the tooling would be delivered as well. Currently, the worker has to walk down to the tooling area, find the tool, and bring it back to his or her work area, which just increases the non-value added time of production.

Poka-yoke activities, which attempt to make it more difficult to make mistakes, also have wide application within Boeing. For example, with the bin system, fixtures that hold the parts together or cutouts that the parts fit in within the bin could be used to enforce inventory accuracy. If too few or too many parts were present, it would be immediately obvious.

Development of More Accurate Metrics

As Boeing improves its processes, the potential exists for erroneous results to be reported because of the method that Boeing uses to track system performance. One example of this was demonstrated during policy analysis. As process cycle times are reduced, inventory levels will increase unless planning flowtimes are modified accordingly. More robust measures of system performance are needed to evaluate process improvement activities. For example, identifying average component inventory level from raw material release to the first work area until final assembly installation would provide an aggregate measure of inventory level that would decrease as process cycle times were improved. Other measures such as lead time to work content would identify how much queue time still exists within a process flow. The important point is that, in many cases, global metrics are needed to avoid confusion. Attempting to determine system-wide performance based on bin inventory levels in one area will not provide much insight.

Systems Approach to Process Improvement

As Boeing attempts to improve its processes, a system's approach should be used. Local optimums will not save the company money in the long run. Aircraft manufacture, from raw material delivery to aircraft roll out, should be considered to be part of one production system, and it is this system that should be optimized. As Boeing transitions to become a lean manufacturer, linkages between its "production islands" need to be tightened, and new organizational and production structures, which eliminate these linkages altogether, need to be developed.[1]

---

[1] Kevin Bartelson, Class of 1993 Leaders For Manufacturing Fellow and Boeing employee, is leading the way in developing one such structure. Please refer to his thesis entitled "Analysis and Redesign of the Production System Structure for 757 and 737 Aircraft Doors".

To aid in this effort, process charts like the one identified in Figure 6.2 need to be developed for each product. On this chart, all pertinent process information for a given component would reside. The actual "touch time" for the part would be determined along with typical queue and wait times during production. Based on delivery performance, process variability could be characterized. Planning flowtimes could then be designed to encompass the $+3\sigma$ limits of the process. Order Point would be a function of the planning flowtime requirement and the current assembly time span order quantity.

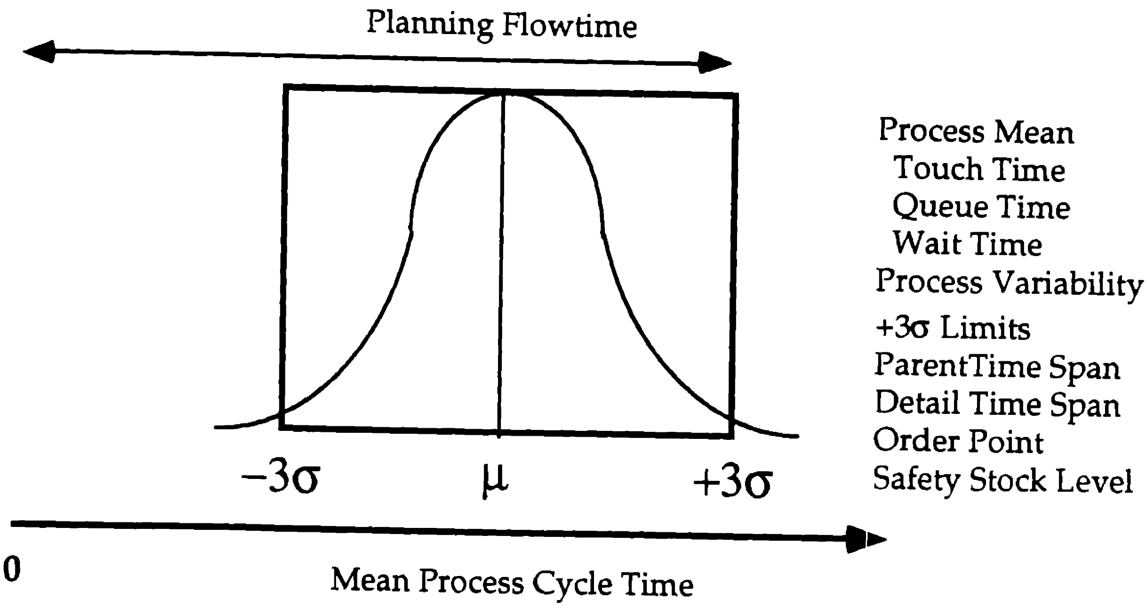


Figure 6.2 - Process Chart

For example, if the parent assembly is built in lot sizes corresponding to a 21 day time span, and the detail process $\mu = 25$ days and $\sigma = 5$ days, the detail would need to be produced using a 3 bin system (63 day time span) with an order point after the second bin was removed. Based on the $+3\sigma$ limits for this detail, 40 days of planning flowtime are required. With a 21 day time span for assembly ordering, ordering the next detail lot when the second bin is removed provides 42 days for detail processing. If the detail process μ = 10 days and σ = 1 days, a two bin system (42 day time span) could be used, with detail orders being released after the second (last) bin was removed.

The important point of this exercise is for the assembly area to develop a better understanding of fabrication process capabilities of internal and external suppliers and order parts based on those capabilities. Currently little knowledge exists of process flowtime requirements. After better metrics a more accurate measurement system are developed, a process chart like the one in Figure 6.2 could be developed and would help assembly areas implement a bin system for ordering parts.

Areas for Further Research

Continuing Evolution of Model Structure and Parameters

It is my hope that the model will see several more iterations which will allow it to become an active tool for policy analysis at Boeing. Specific areas for possible improvement include : 1) improvement of doubler structure and/or parameter values to improve Mean Absolute Percent Error, 2) incorporation of engineering change perturbative effects on ordering action, 3) broadening the measure of expedite and shortage impact, 4) improvements in the expediter decision logic to split orders, 5) representation of priority cycles and impact on order queue time within production areas, and 6) inclusion of financial measures of expedite action, shortages, product losses, and inventory holding costs.

Development of Business System Architecture for Product Centers

It is likely that a transition from the current "Legacy" systems to a new business system architecture will start with the Product Centers. Ample opportunity exists for research in this area. What should the business systems used by Boeing's Product Centers look like ? What types of information are needed, and maybe more importantly what types of information are not ? What hardware and software platforms should be used ? Should a centralized, decentralized, or hybrid organizational structure be used to manage these new business systems ?

Specific Implementation of Bin System

The bin system discussed in this thesis provides an opportunity to create a stronger link between detail and assembly production. A pilot program to demonstrate the bin system concept would be a good next step. Details and assemblies that constitute a product family could be used for the pilot. For example, in Renton Lot Time, all details and assemblies that support the Door Product Center might be produced under the bin system concept.

References

Compton & Heim: Manufacturing Systems: Foundations of World-Class Practice, National Academy Press: Washington D.C., 1992.

Copes: "Use of a Simple Queueing and Systems Model to Evaluate Factory Flowtime Performance", MIT Thesis, 1992.

D'Azzo, et al: Linear Control System Analysis and Design, McGraw-Hill: New York, 1981.

Deming: Out of the Crisis, MIT Center for Advanced Engineering Study: Cambridge, 1982.

Dertouzos, et al: Made in America, MIT Press: Cambridge, 1989.

Forrester & Senge: "Test for Building Confidence in System Dynamics Models", TIMS Studies in the Management Sciences, 14, pp. 209-228.

Forrester: Industrial Dynamics, Productivity Press: Cambridge, 1961.

High Performance Systems. Ithink User's Guide, Hanover, NH: High Performance Systems Inc.

Lyneis: Corporate Planning and Policy Design, Pugh-Roberts Inc.: Cambridge, 1988.

Merli: Total Manufacturing Management, Productivity Press: Cambridge, 1990.

MIT Management. "Are Investments in Information Systems paying off ?". An interview with Dean Thurow. Spring 1990. pp.14-16.

Monden: Toyota Production System, Industrial Engineering and Management Press: Norcross, 1983.

Montgomery: Statistical Process Control, John Wiley & Sons: New York, 1991.

Morecroft. "A Systems Perspective on Material Requirements Planning, Decision Sciences, 14, pp. 1-17.

Nahmias: Production and Operations Analysis, Irwin: Boston, 1989.

Ogata: Modern Control Engineering, Prentice Hall, Inc.: Englewood Cliffs, 1970.

Randers: Elements of the System Dynamics Method, Productivity Press: Cambridge, 1980.

Richardson & Pugh: Introduction to System Dynamics Modeling with DYNAMO, Productivity Press: Cambridge, 1981.

Schonberger: World Class Manufacturing Casebook, The Free Press: New York, 1987.

Senge: The Fifth Discipline, Doubleday: New York, 1990.

Shanahan: "Manufacturing System Performance: A Case Analysis of Boeing Sheet Metal Production", MIT Thesis, 1991.

Shingo: A Study of the Toyota Production System, Productivity Press: Cambridge, 1989.

Sterman. "A Skeptic's Guide to Computer Models", University Press of America, pp. 133-169.

Suzaki: The New Manufacturing Challenge, The Free Press: New York, 1987.

Whiting: "Reducing the Flowtime of a Manufacturing Order Release Process", MIT Thesis, 1992.

Womack, et al: The Machine that Changed the World, Rawson Associates: New York, 1990.

Appendix A1 - Brief Tutorial on Model Elements

To aid those unfamiliar with Ithink^{TM} Software, the following tutorial is intended as an aid in reviewing the model diagrams used in this thesis. The model building blocks, shown in Figure A-1.1 (the grouping used has no special meaning and is just being used for instructional purposes), each have special features. The information for this tutorial is taken from the Ithink^{TM} User's Manual.

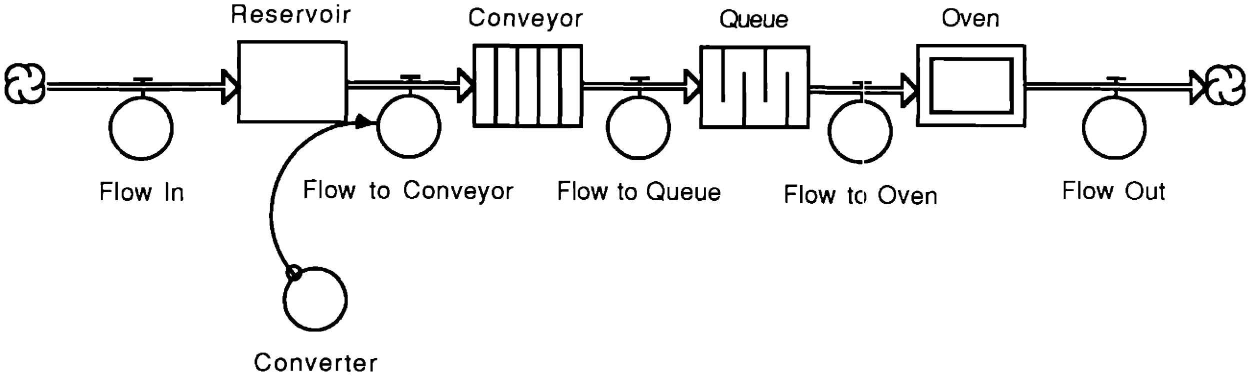


Figure A-1.1 - Example of Model Elements

Four Types of Stocks

Reservoir:
units enter and "pile up" into a heap, losing individual identity.

Conveyor:
attaches via outflow to a reservoir. A conveyor is like a "moving sidewalk"; units get on, ride for a fixed amount of time, and then exit.

Queue:
attaches via outflow to an oven. If Oven doors are closed, a line grows. The line is a Queue.

Oven:
Inflow is outflow from a queue. Oven reaches back into queue an amount of time = "bunching depth" and then admits units, if: oven is not filled to capacity, and the "fill time" has not been exceeded, and the oven is not "baking". The oven then "bakes" units for a specified amount of time called the "cook time".

Flows

A flow empties into or drains out of a stock. The rate at which this occurs is dependent on the formulation of the equation that is embedded within the flow. The flow can be a constant, or it can be a more complicated function dependent on several different variables.

Converters

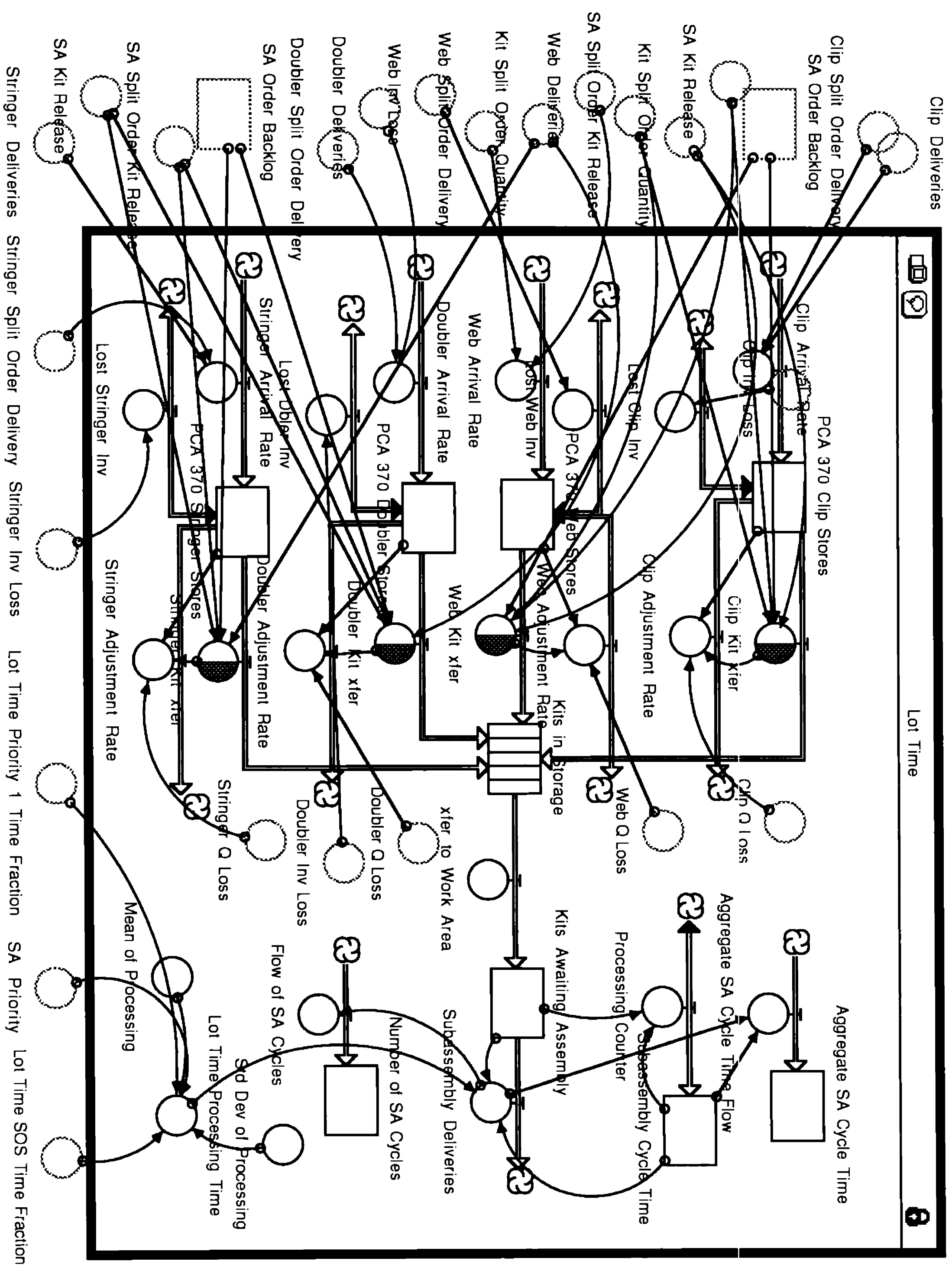
A converter can be a constant, a built-in function, a user created algebraic equation, or a graphical function. A converter can feed information to flows, or it can be used as a stock substitute.

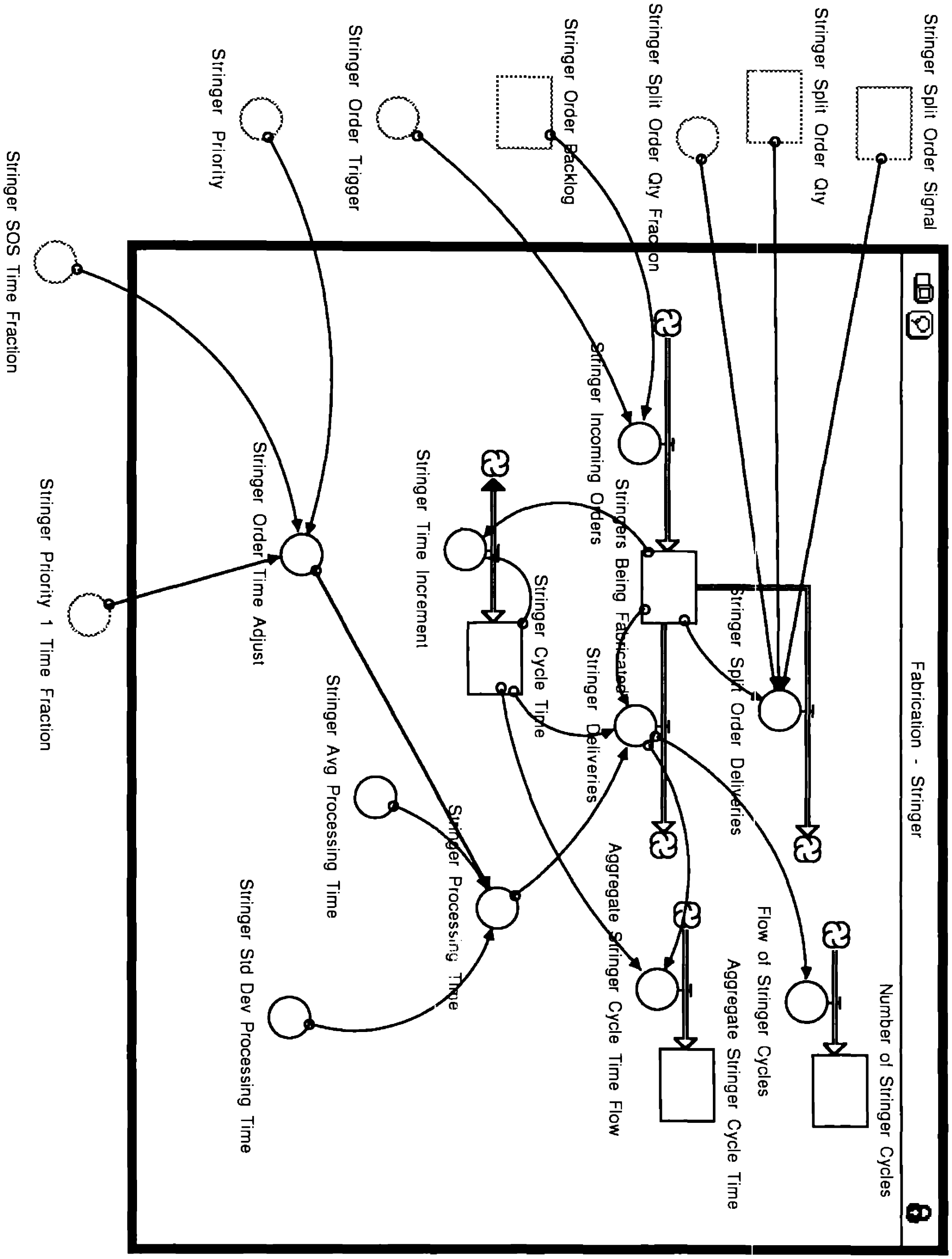
Connector

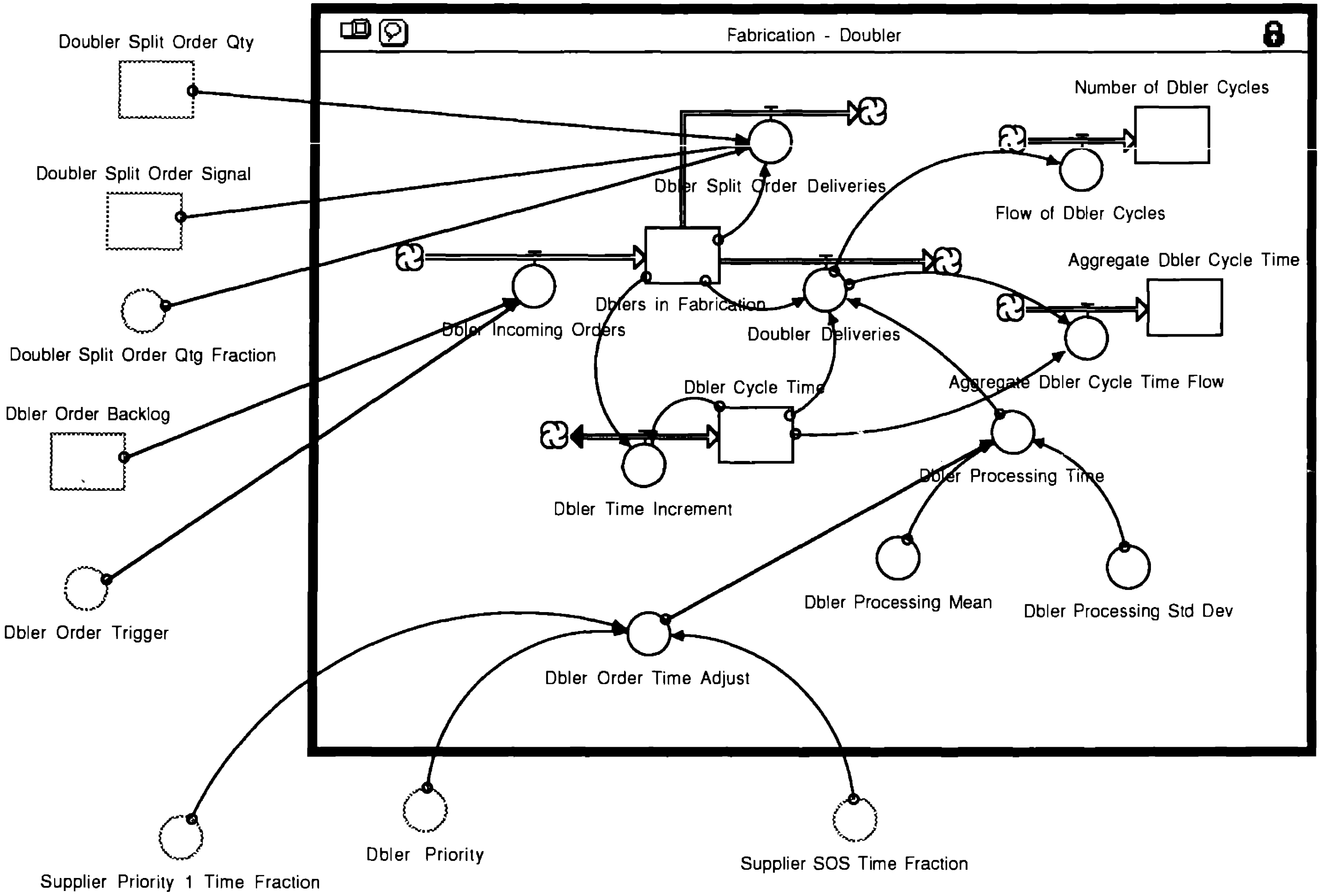
A connector links two entities together and is represented by the arrow linkage that ties the converter to the flow in Figure A-1.1. It allows the information available at one point to be made available at other locations within the model. It is the mechanism that allows for feedback between different portions of the model structure.

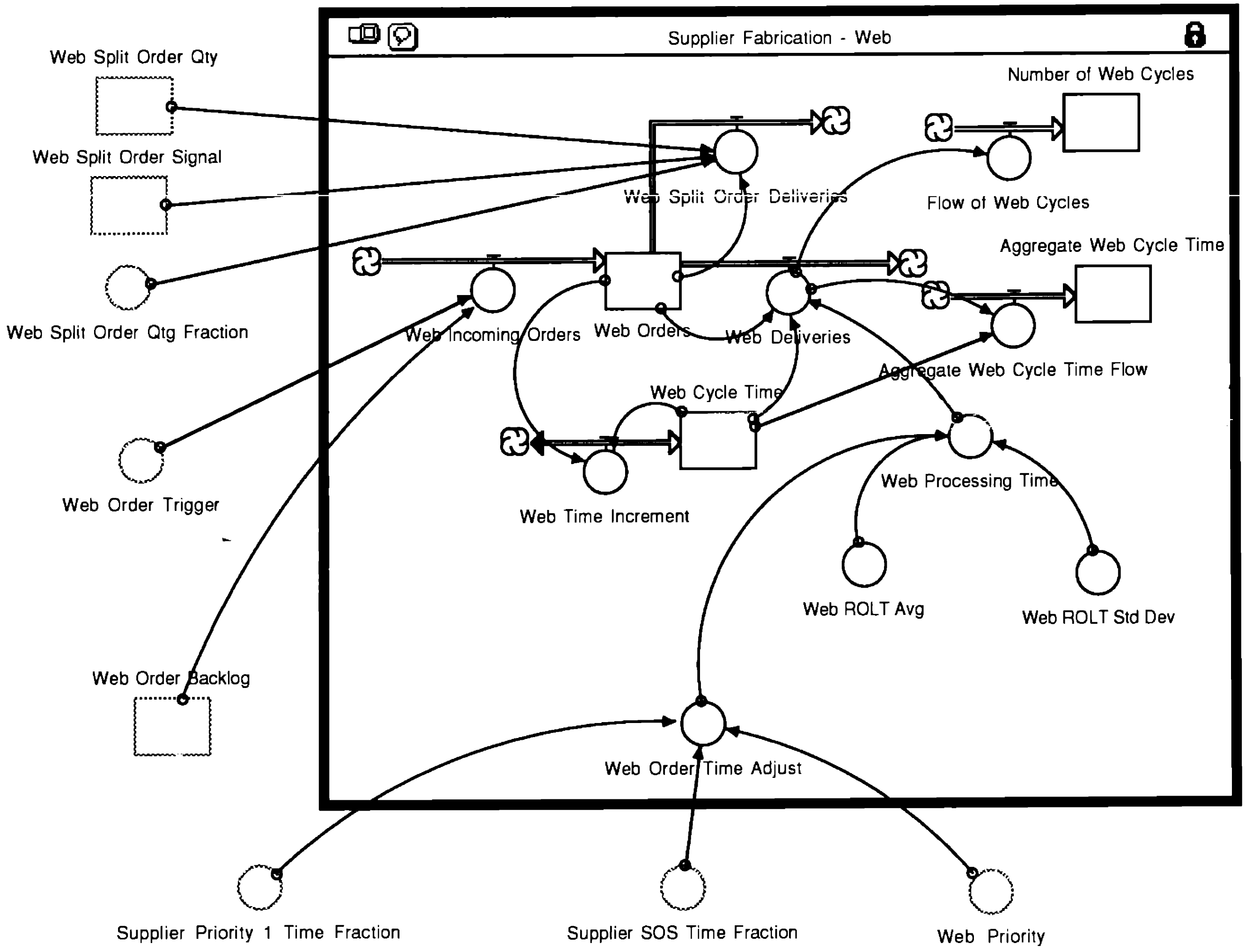
For a more detailed discussion of Ithink software and how it is used, please refer to the Ithink™ User's Manual produced by High Performance Systems, Inc. .

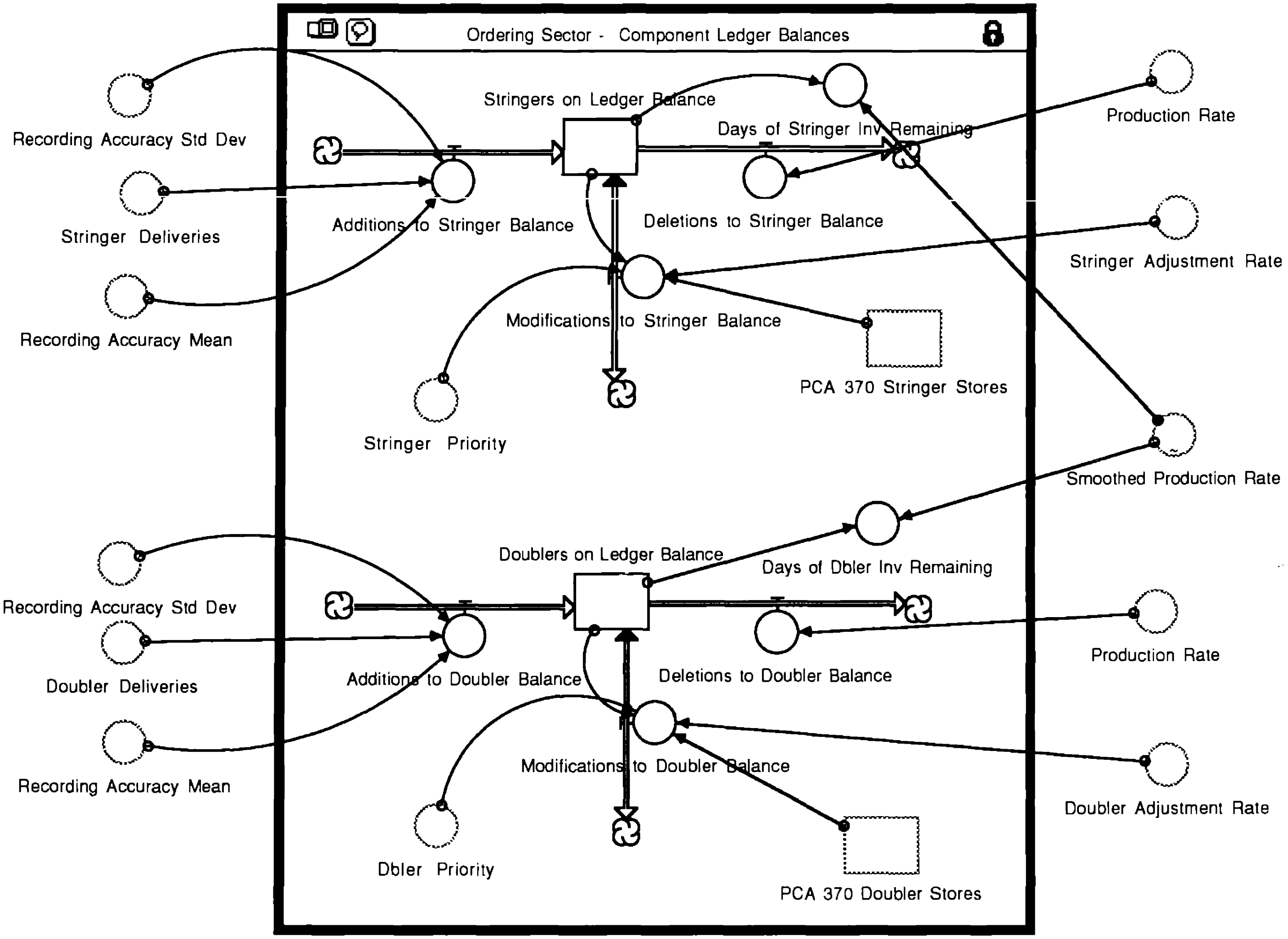
Appendix A2 - Model Diagrams



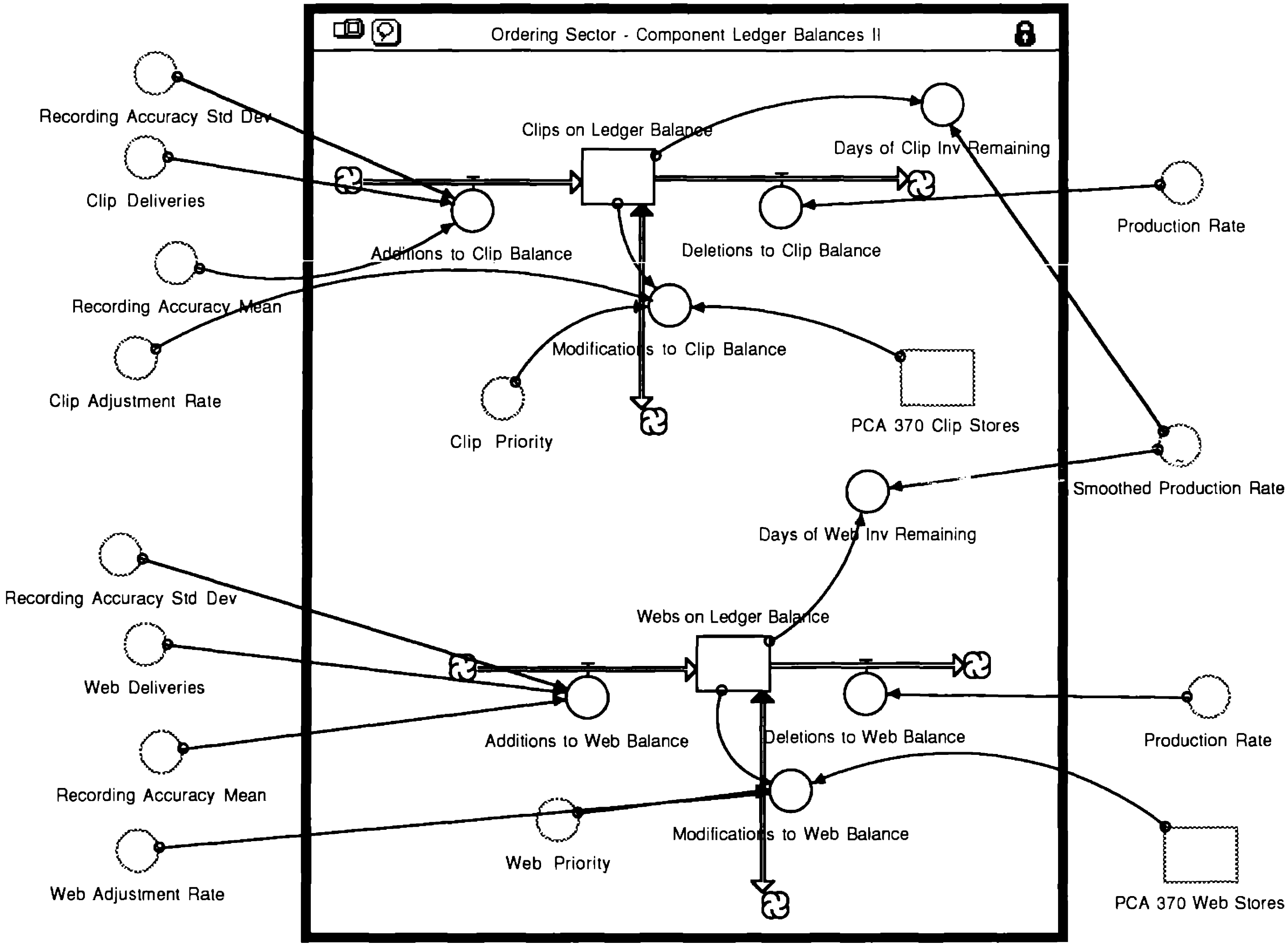

Fabrication - Stringer
Stringer Split Order Signal
Stringer Split Order Qty
Stringer Split Order Qty Fraction
Stringer Order Backlog
Stringer Order Trigger
Stringer Priority
Stringer SOS Time Fraction
Stringer Incoming Orders
Stringers Being Fabricated
Stringer Split Order Deliveries
Stringer Deliveries
Stringer Time Increment
Stringer Cycle Time
Stringer Order Time Adjust
Stringer Priority 1 Time Fraction
Number of Stringer Cycles
Flow of Stringer Cycles
Aggregate Stringer Cycle Time
Aggregate Stringer Cycle Time Flow
Stringer Processing Time
Stringer Avg Processing Time
Stringer Std Dev Processing Time

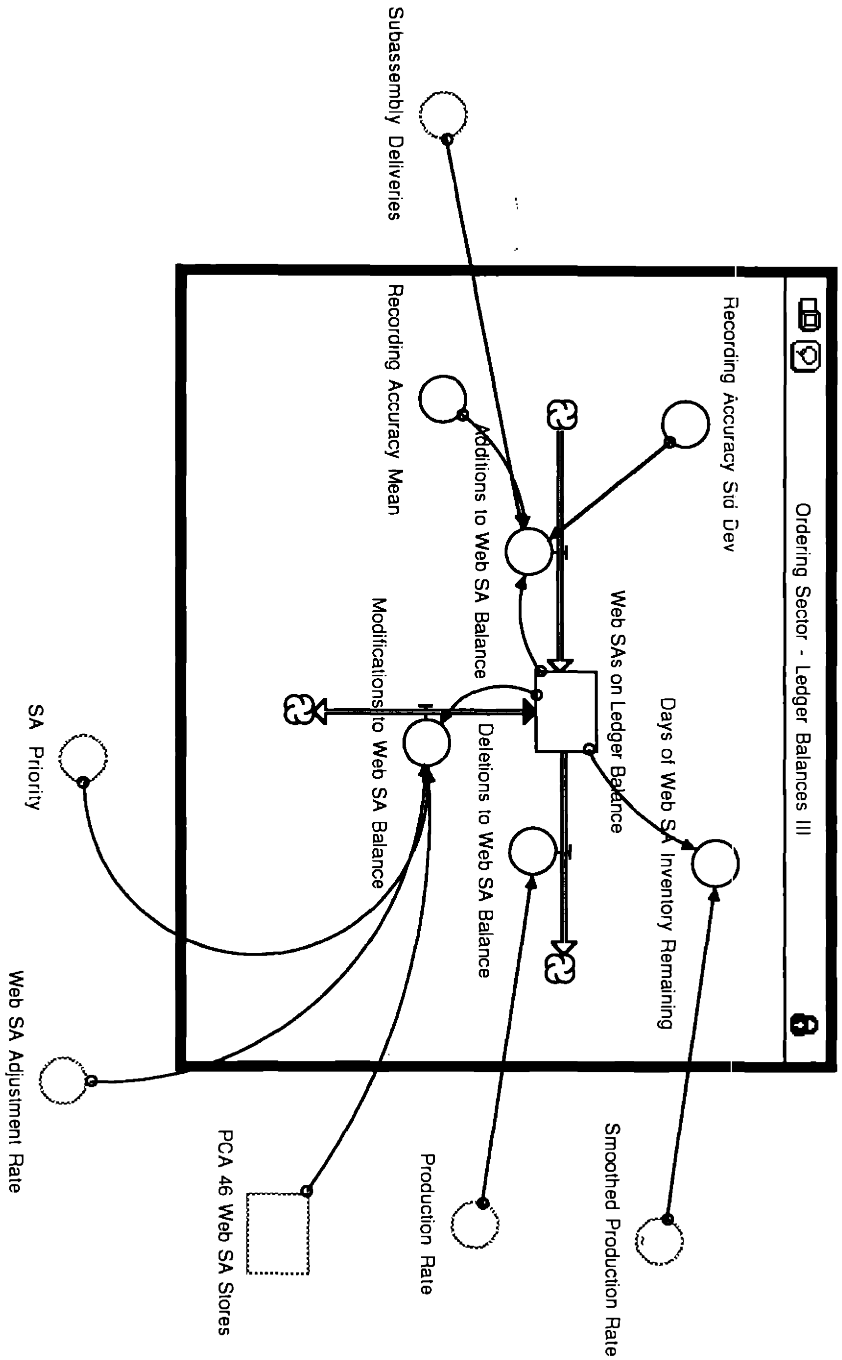

Fabrication - Doubler
Doubler Split Order Qty
Doubler Split Order Signal
Doubler Split Order Qtg Fraction
Dbler Order Backlog
Dbler Order Trigger
Number of Dbler Cycles
Dbler Split Order Deliveries
Flow of Dbler Cycles
Aggregate Dbler Cycle Time
Dblers in Fabrication
Dbler Incoming Orders
Doubler Deliveries
Aggregate Dbler Cycle Time Flow
Dbler Cycle Time
Dbler Processing Time
Dbler Time Increment
Dbler Processing Mean
Dbler Processing Std Dev
Dbler Order Time Adjust
Supplier Priority 1 Time Fraction
Dbler Priority
Supplier SOS Time Fraction

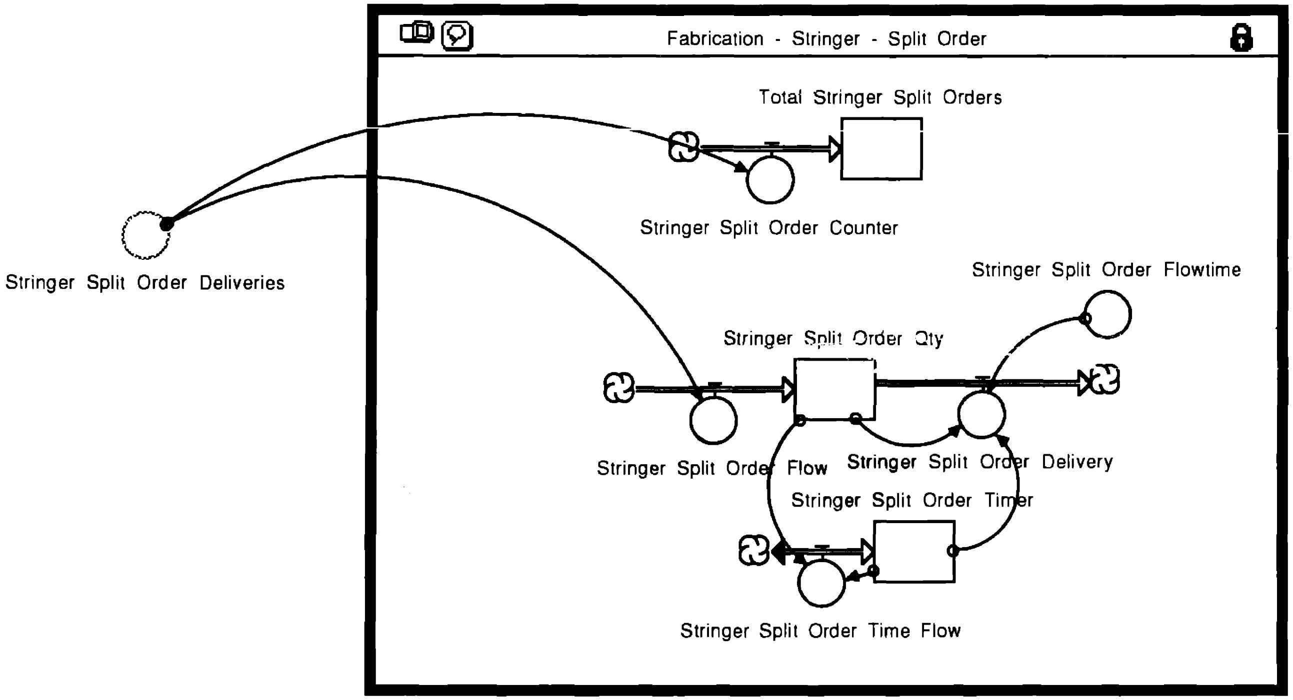

Supplier Fabrication - Web
Web Split Order Qty
Web Split Order Signal
Web Split Order Qtg Fraction
Web Order Trigger
Web Order Backlog
Number of Web Cycles
Web Split Order Deliveries
Flow of Web Cycles
Aggregate Web Cycle Time
Web Incoming Orders
Web Orders
Web Deliveries
Aggregate Web Cycle Time Flow
Web Cycle Time
Web Time Increment
Web Processing Time
Web ROLT Avg
Web ROLT Std Dev
Web Order Time Adjust
Supplier Priority 1 Time Fraction
Supplier SOS Time Fraction
Web Priority

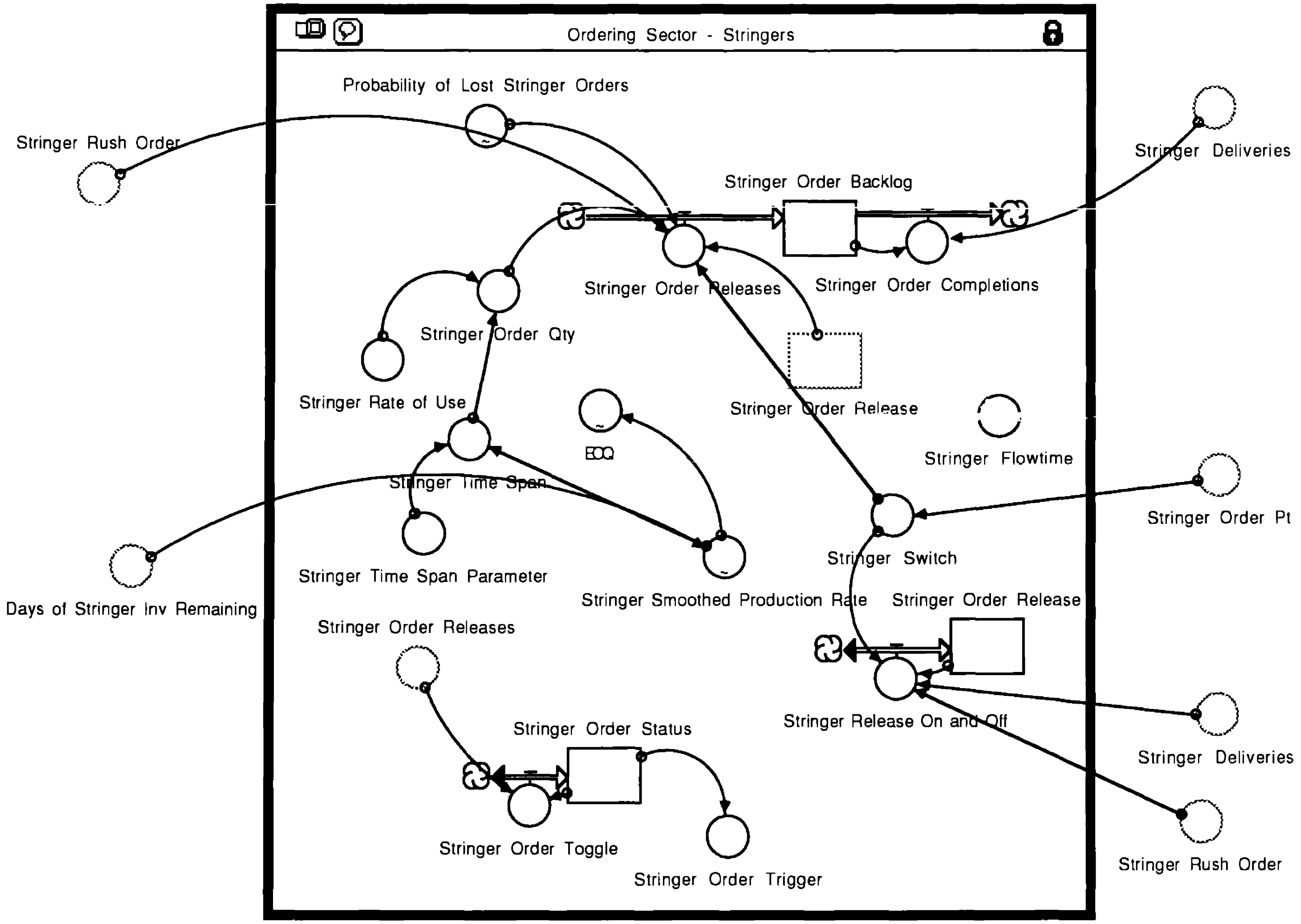


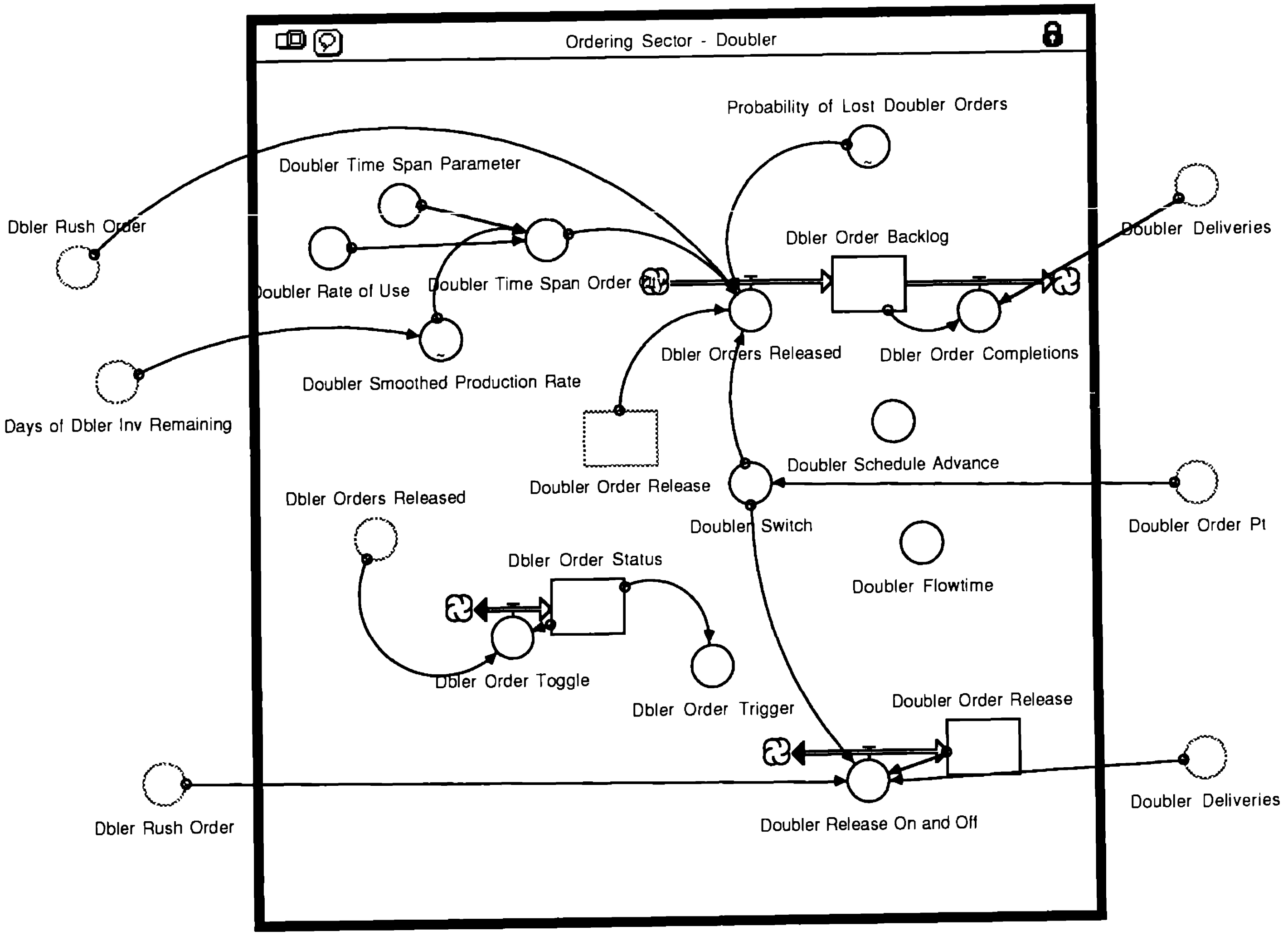
Ordering Sector - Component Ledger Balances
Recording Accuracy Std Dev
Stringer Deliveries
Recording Accuracy Mean
Additions to Stringer Balance
Stringers on Ledger Balance
Days of Stringer Inv Remaining
Deletions to Stringer Balance
Modifications to Stringer Balance
Stringer Priority
PCA 370 Stringer Stores
Production Rate
Stringer Adjustment Rate
Smoothed Production Rate
Recording Accuracy Std Dev
Doubler Deliveries
Recording Accuracy Mean
Additions to Doubler Balance
Doublers on Ledger Balance
Days of Dbler Inv Remaining
Deletions to Doubler Balance
Modifications to Doubler Balance
Dbler Priority
PCA 370 Doubler Stores
Production Rate
Doubler Adjustment Rate

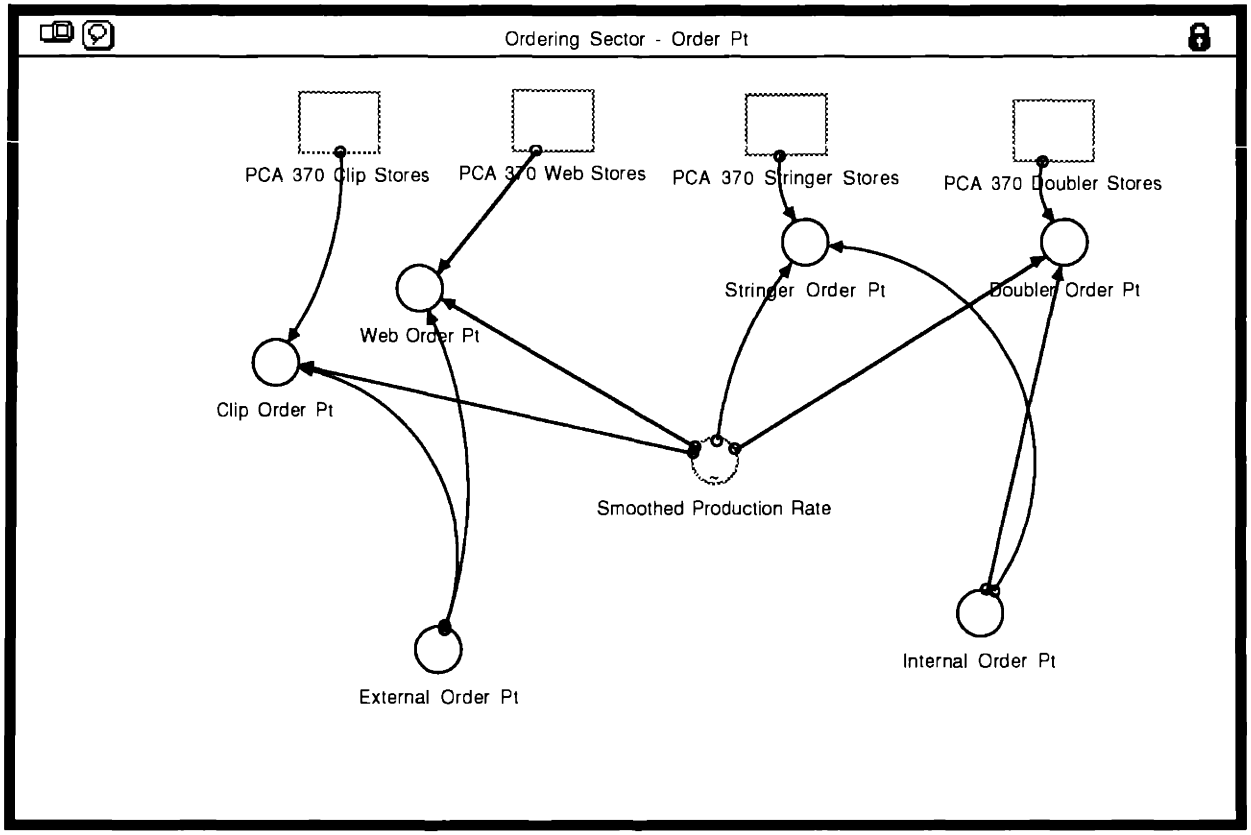

Ordering Sector - Component Ledger Balances II
Recording Accuracy Std Dev
Clip Deliveries
Recording Accuracy Mean
Clip Adjustment Rate
Clips on Ledger Balance
Additions to Clip Balance
Deletions to Clip Balance
Days of Clip Inv Remaining
Modifications to Clip Balance
Clip Priority
PCA 370 Clip Stores
Production Rate
Smoothed Production Rate
Days of Web Inv Remaining
Recording Accuracy Std Dev
Web Deliveries
Recording Accuracy Mean
Web Adjustment Rate
Webs on Ledger Balance
Additions to Web Balance
Deletions to Web Balance
Modifications to Web Balance
Web Priority
Production Rate
PCA 370 Web Stores

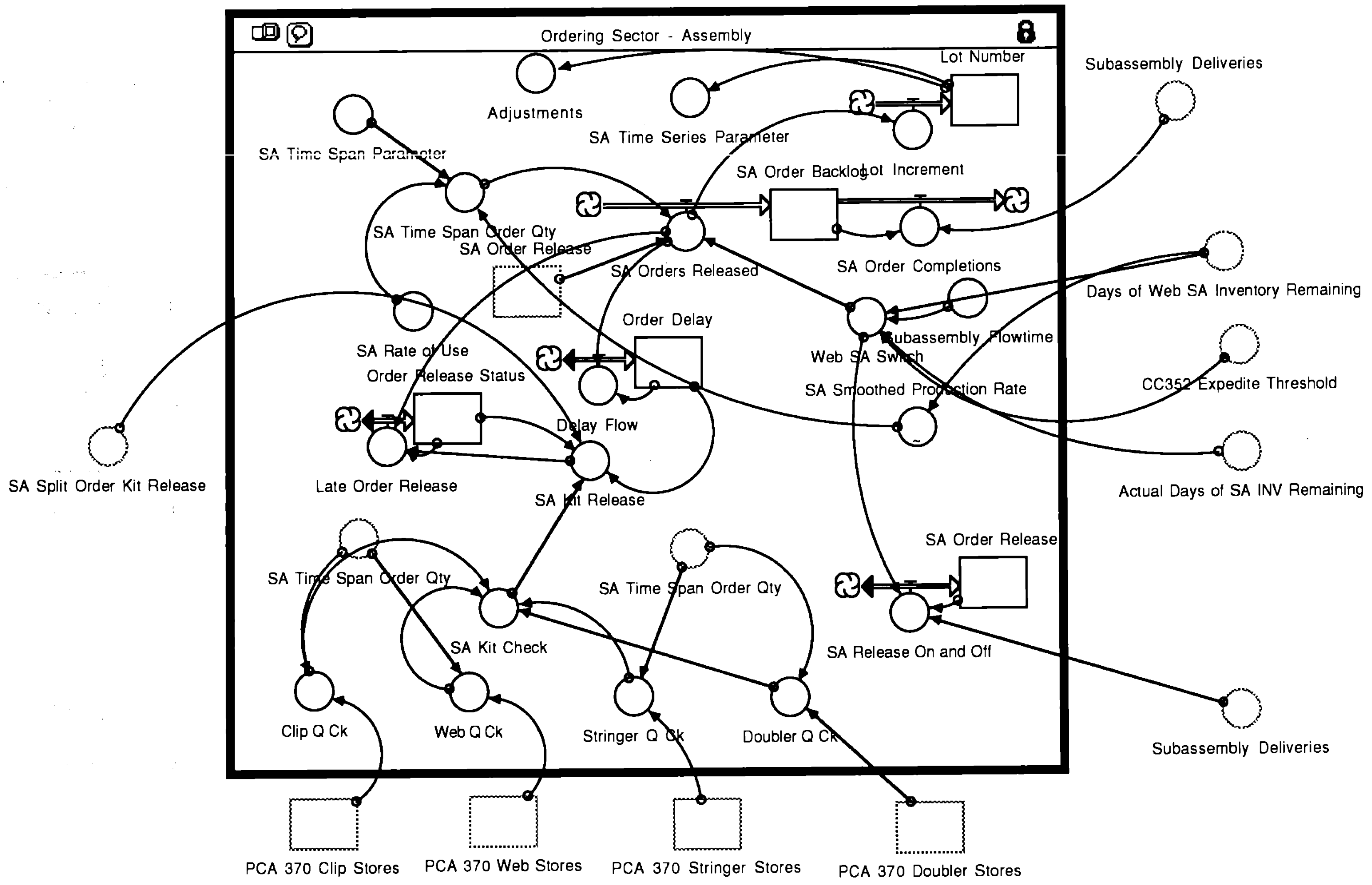

Ordering Sector - Ledger Balances III
Recording Accuracy Std Dev
Recording Accuracy Mean
Subassembly Deliveries
Additions to Web SA Balance
Web SAs on Ledger Balance
Days of Web SA Inventory Remaining
Deletions to Web SA Balance
Modifications to Web SA Balance
SA Priority
Web SA Adjustment Rate
PCA 46 Web SA Stores
Production Rate
Smoothed Production Rate

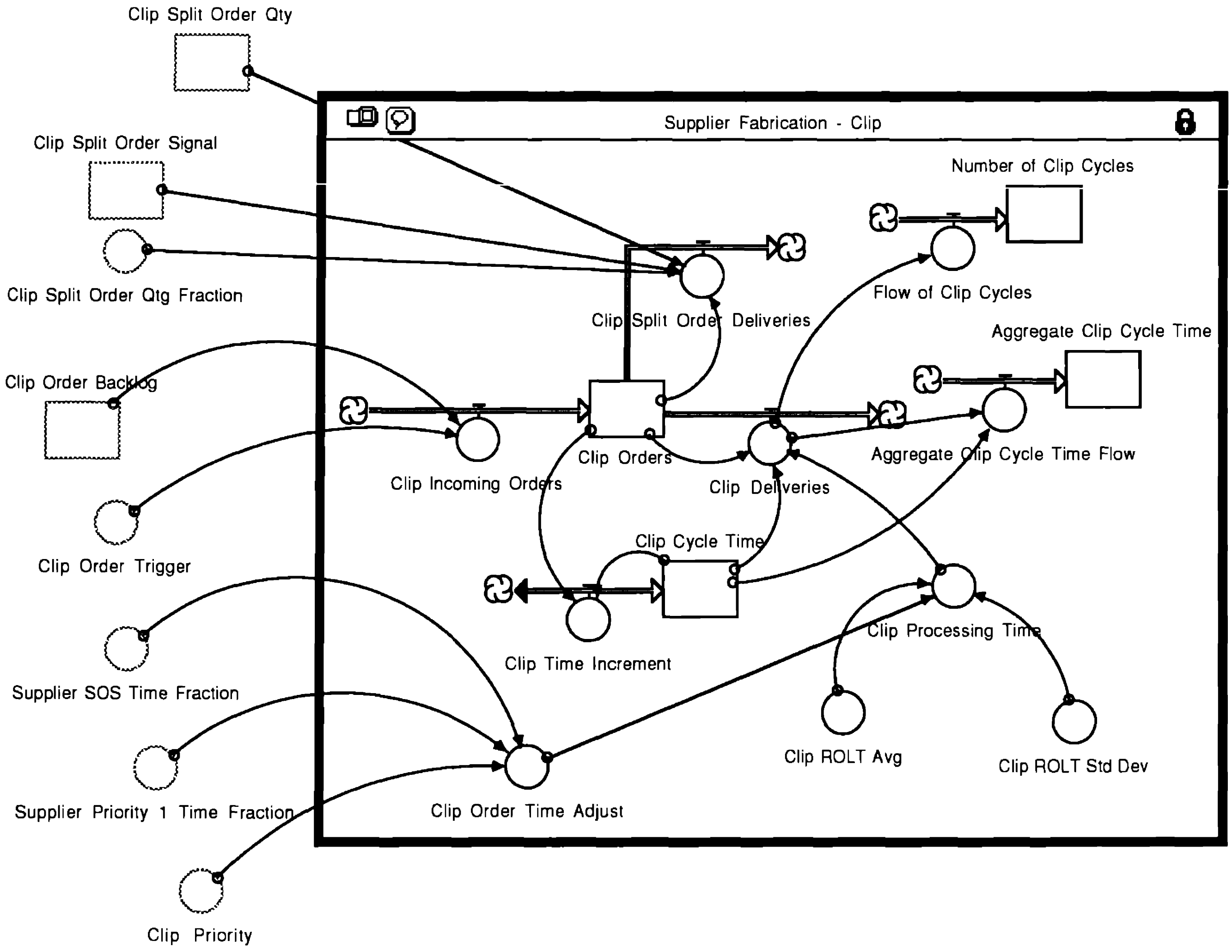

Fabrication - Stringer - Split Order
Total Stringer Split Orders
Stringer Split Order Counter
Stringer Split Order Deliveries
Stringer Split Order Flowtime
Stringer Split Order Qty
Stringer Split Order Flow
Stringer Split Order Delivery
Stringer Split Order Timer
Stringer Split Order Time Flow

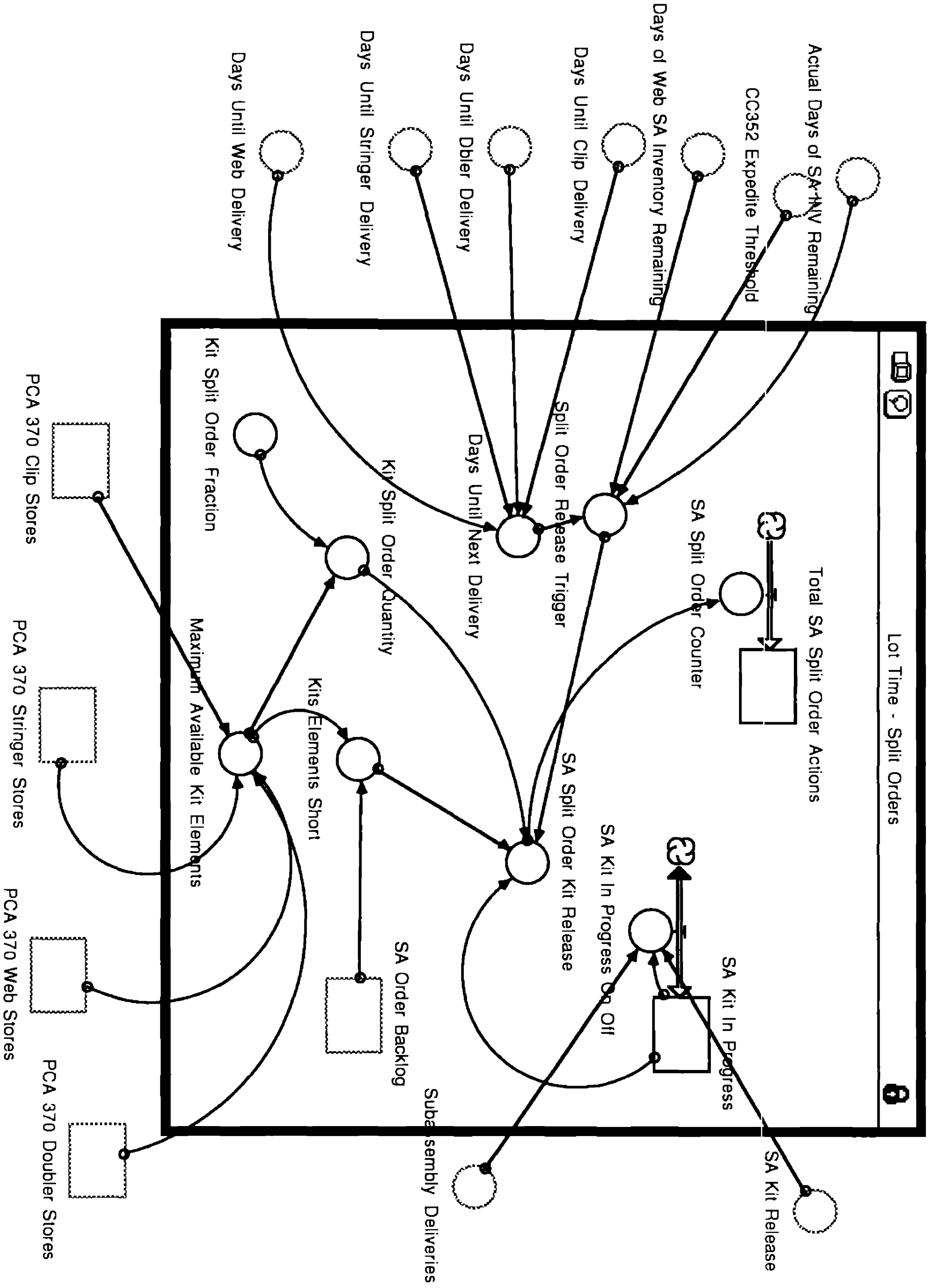

Ordering Sector - Stringers
Probability of Lost Stringer Orders
Stringer Rush Order
Stringer Order Backlog
Stringer Deliveries
Stringer Order Releases
Stringer Order Completions
Stringer Order Qty
Stringer Rate of Use
Stringer Order Release
EOQ
Stringer Flowtime
Stringer Time Span
Stringer Time Span Parameter
Stringer Order Pt
Stringer Switch
Days of Stringer Inv Remaining
Stringer Smoothed Production Rate
Stringer Order Release
Stringer Order Releases
Stringer Release On and Off
Stringer Deliveries
Stringer Order Status
Stringer Order Toggle
Stringer Order Trigger
Stringer Rush Order

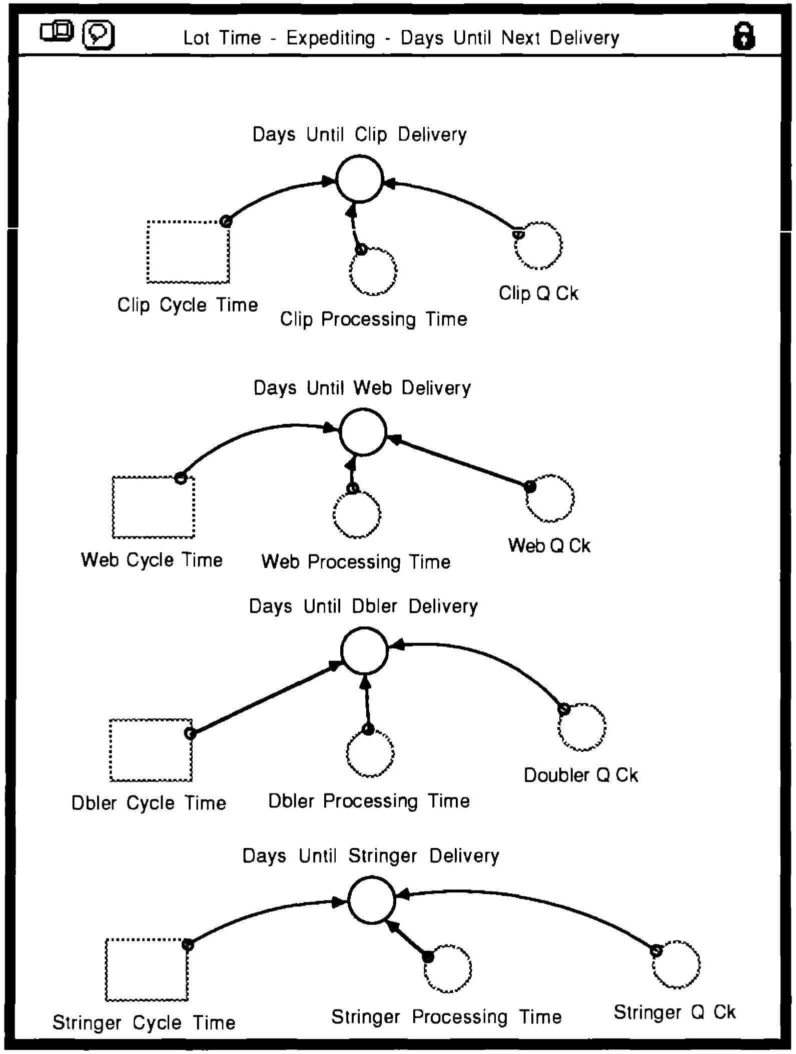

Ordering Sector - Doubler
Probability of Lost Doubler Orders
Doubler Time Span Parameter
Dbler Rush Order
Dbler Order Backlog
Doubler Deliveries
oubler Rate of Use
Doubler Time Span Order Qty
Dbler Orders Released
Dbler Order Completions
Days of Dbler Inv Remaining
Doubler Smoothed Production Rate
Doubler Schedule Advance
Doubler Order Release
Dbler Orders Released
Doubler Switch
Doubler Order Pt
Dbler Order Status
Doubler Flowtime
Dbler Order Toggle
Dbler Order Trigger
Doubler Order Release
Dbler Rush Order
Doubler Release On and Off
Doubler Deliveries

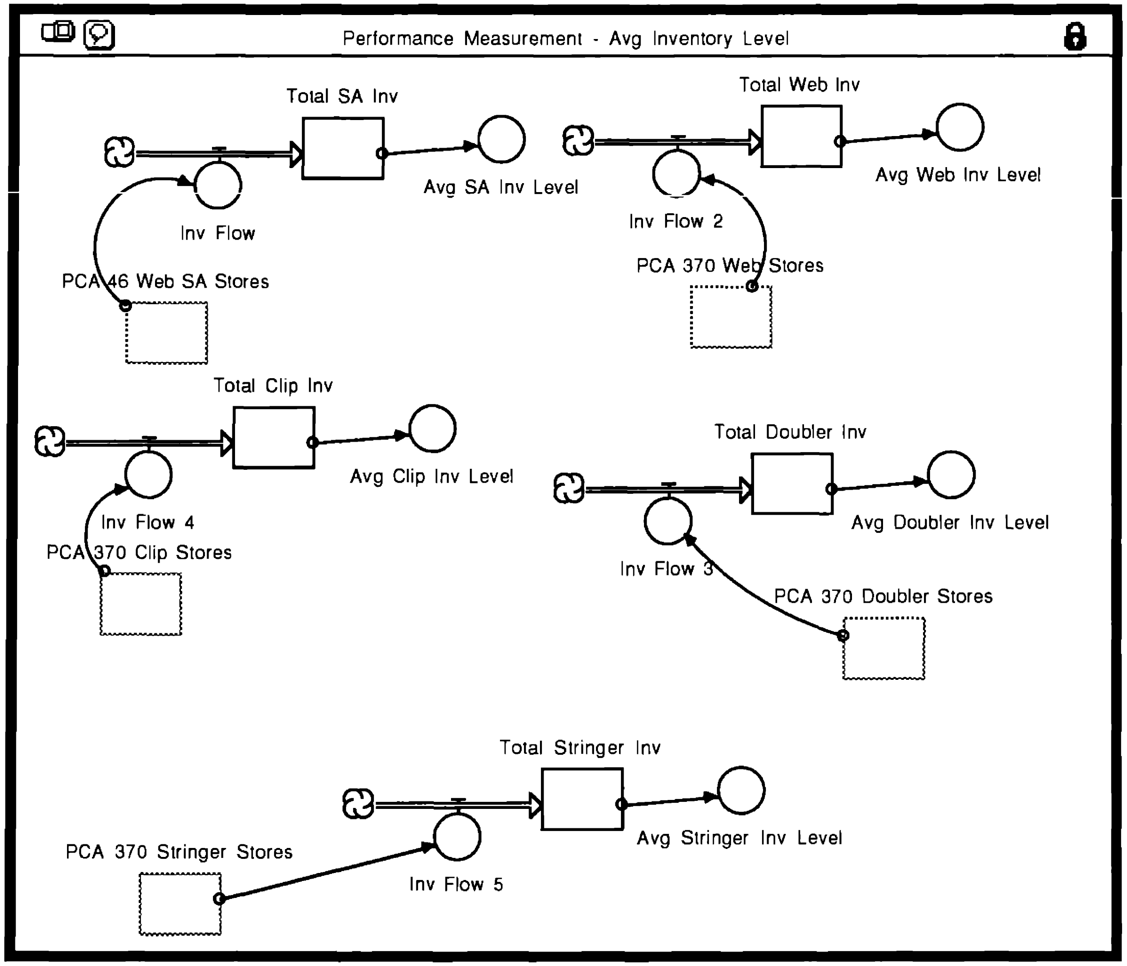

Ordering Sector - Order Pt
PCA 370 Clip Stores
PCA 370 Web Stores
PCA 370 Stringer Stores
PCA 370 Doubler Stores
Stringer Order Pt
Doubler Order Pt
Web Order Pt
Clip Order Pt
Smoothed Production Rate
Internal Order Pt
External Order Pt

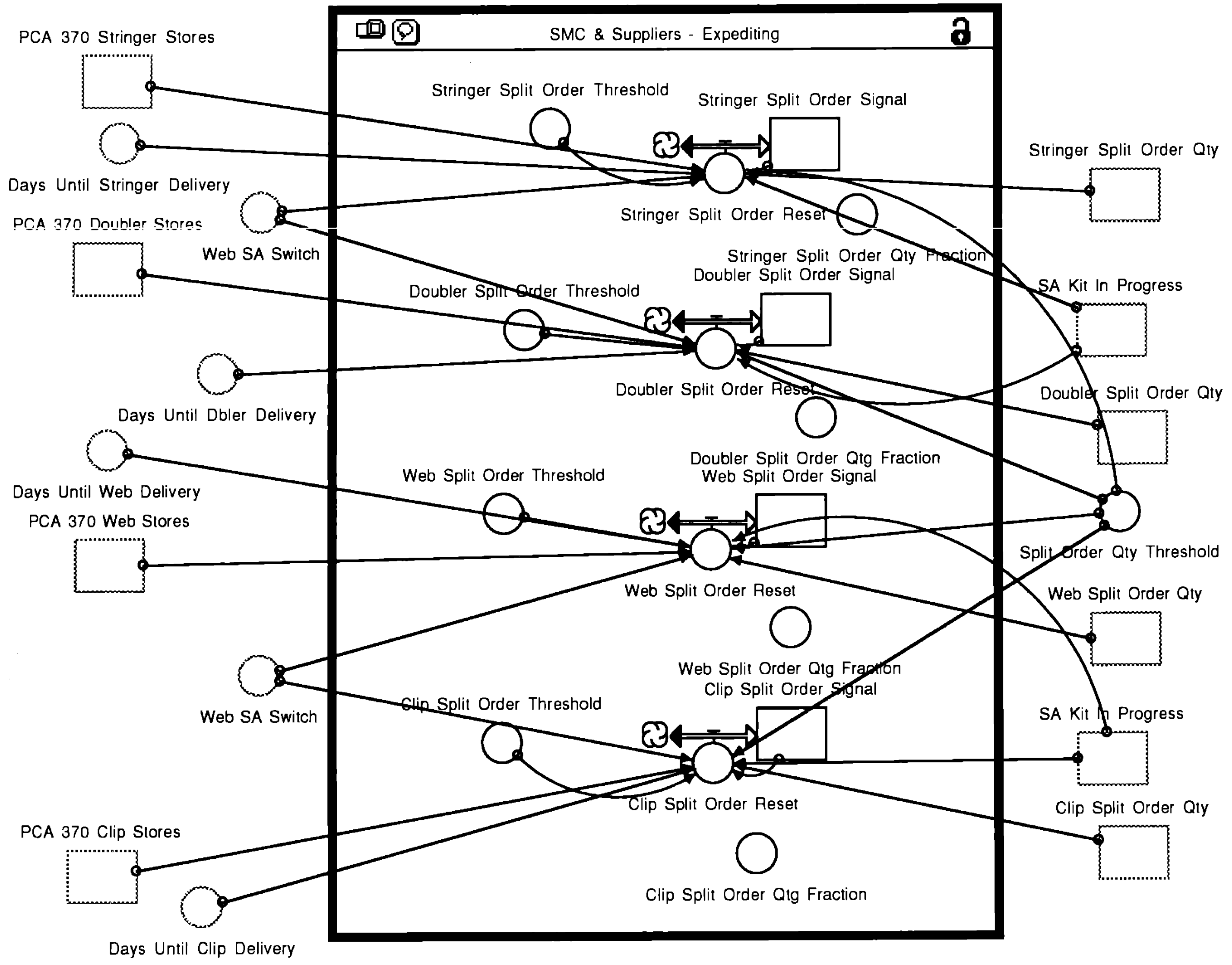

Ordering Sector - Assembly
Adjustments
SA Time Series Parameter
Lot Number
Subassembly Deliveries
SA Time Span Parameter
Lot Increment
SA Order Backlog
SA Time Span Order Qty
SA Order Release
SA Orders Released
SA Order Completions
Days of Web SA Inventory Remaining
Order Delay
Subassembly Flowtime
Web SA Switch
SA Rate of Use
Order Release Status
SA Smoothed Production Rate
CC352 Expedite Threshold
Delay Flow
SA Split Order Kit Release
Late Order Release
SA Kit Release
Actual Days of SA INV Remaining
SA Order Release
SA Time Span Order Qty
SA Time Span Order Qty
SA Kit Check
SA Release On and Off
Clip Q Ck
Web Q Ck
Stringer Q Ck
Doubler Q Ck
Subassembly Deliveries
PCA 370 Clip Stores
PCA 370 Web Stores
PCA 370 Stringer Stores
PCA 370 Doubler Stores

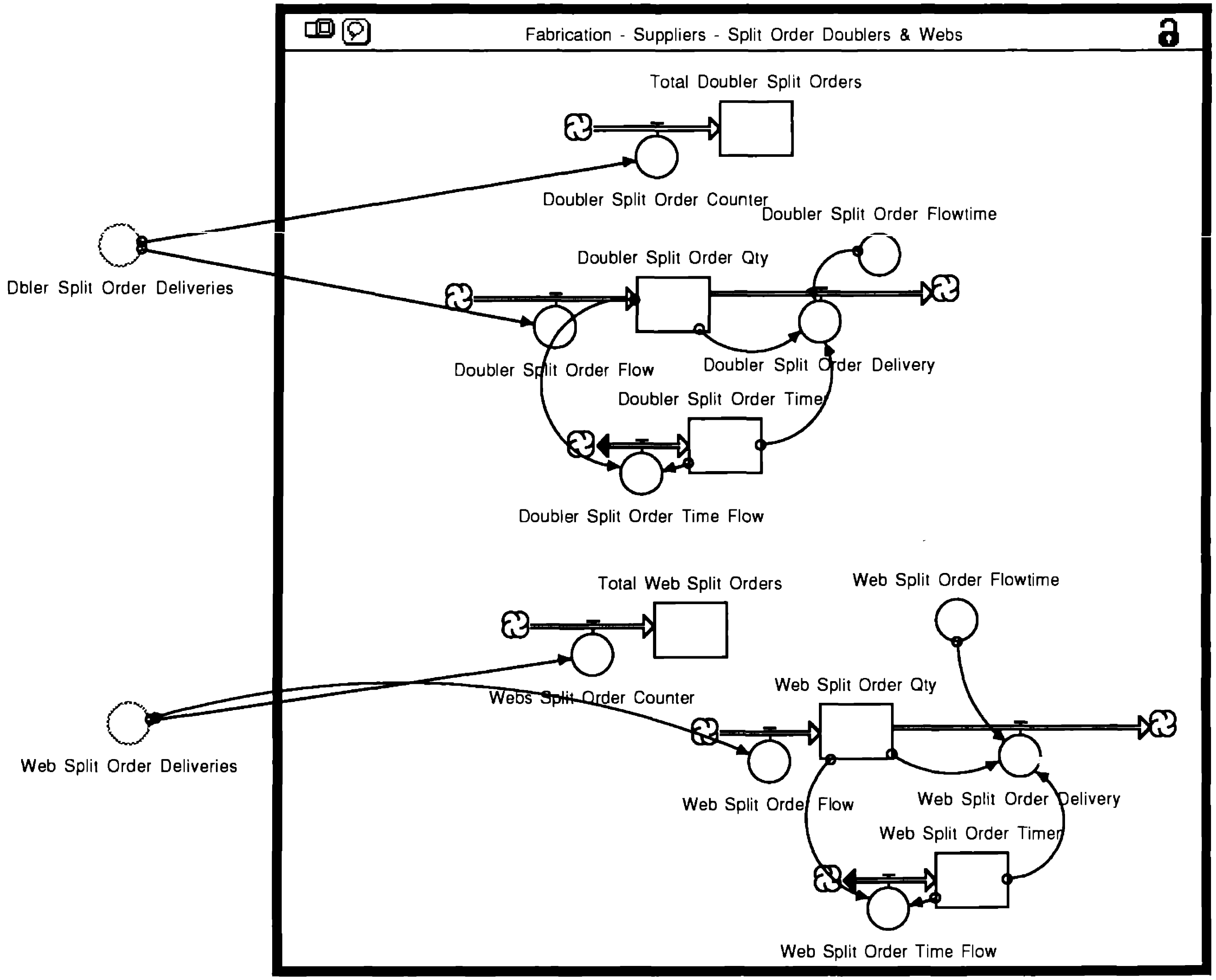

Supplier Fabrication - Clip
Clip Split Order Qty
Clip Split Order Signal
Clip Split Order Qtg Fraction
Clip Order Backlog
Clip Order Trigger
Supplier SOS Time Fraction
Supplier Priority 1 Time Fraction
Clip Priority
Number of Clip Cycles
Flow of Clip Cycles
Clip Split Order Deliveries
Aggregate Clip Cycle Time
Clip Incoming Orders
Clip Orders
Clip Deliveries
Aggregate Clip Cycle Time Flow
Clip Cycle Time
Clip Time Increment
Clip Processing Time
Clip ROLT Avg
Clip ROLT Std Dev
Clip Order Time Adjust

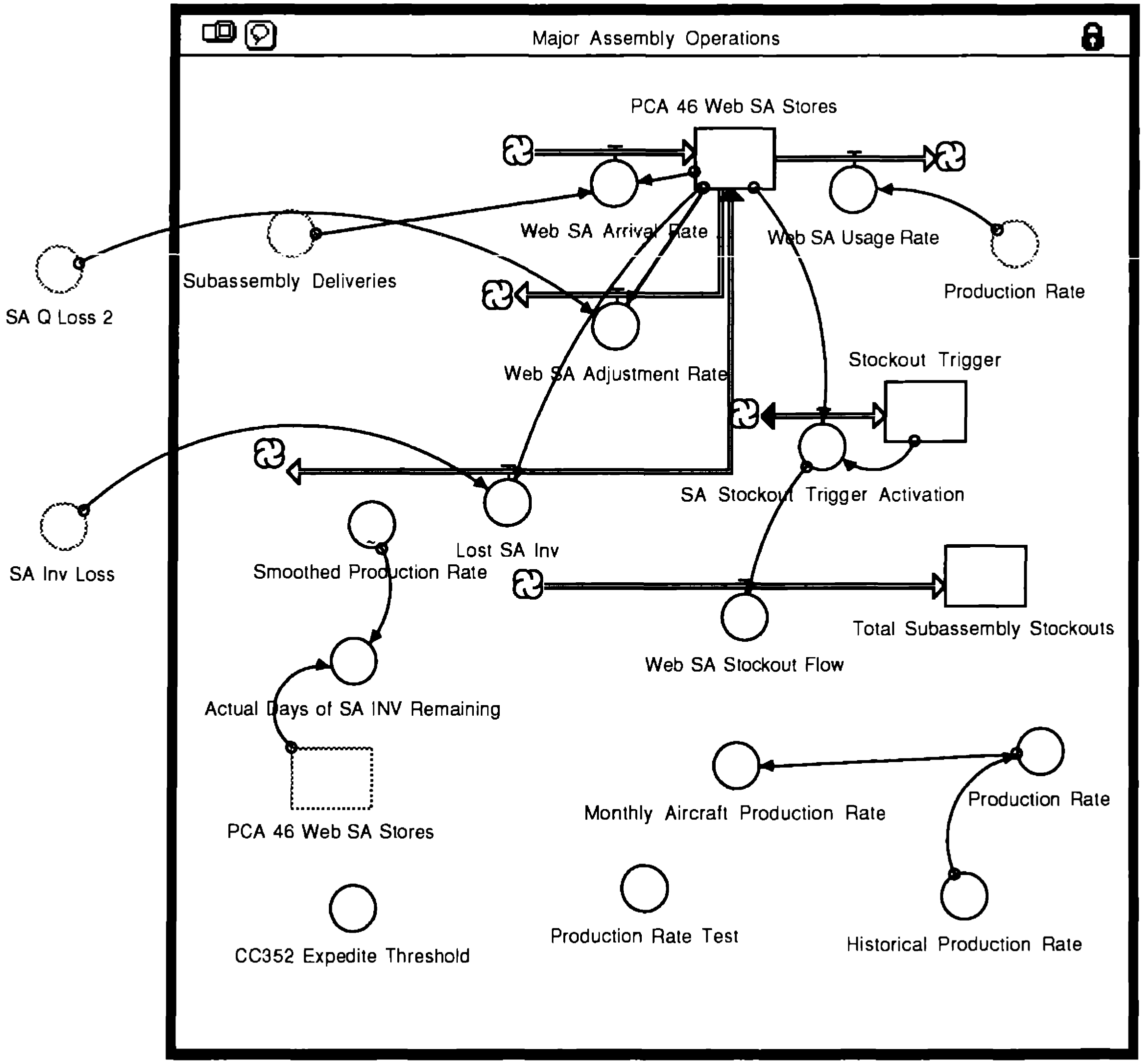

Lot Time - Split Orders
Total SA Split Order Actions
SA Split Order Counter
Split Order Release Trigger
Days Until Next Delivery
Kit Split Order Quantity
Kit Split Order Fraction
Kits Elements Short
Maximum Available Kit Elements
SA Split Order Kit Release
SA Kit In Progress On Off
SA Kit In Progress
SA Order Backlog
Actual Days of SA INV Remaining
CC352 Expedite Threshold
Days of Web SA Inventory Remaining
Days Until Clip Delivery
Days Until Dbler Delivery
Days Until Stringer Delivery
Days Until Web Delivery
PCA 370 Clip Stores
PCA 370 Stringer Stores
PCA 370 Web Stores
PCA 370 Doubler Stores
Subassembly Deliveries
SA Kit Release

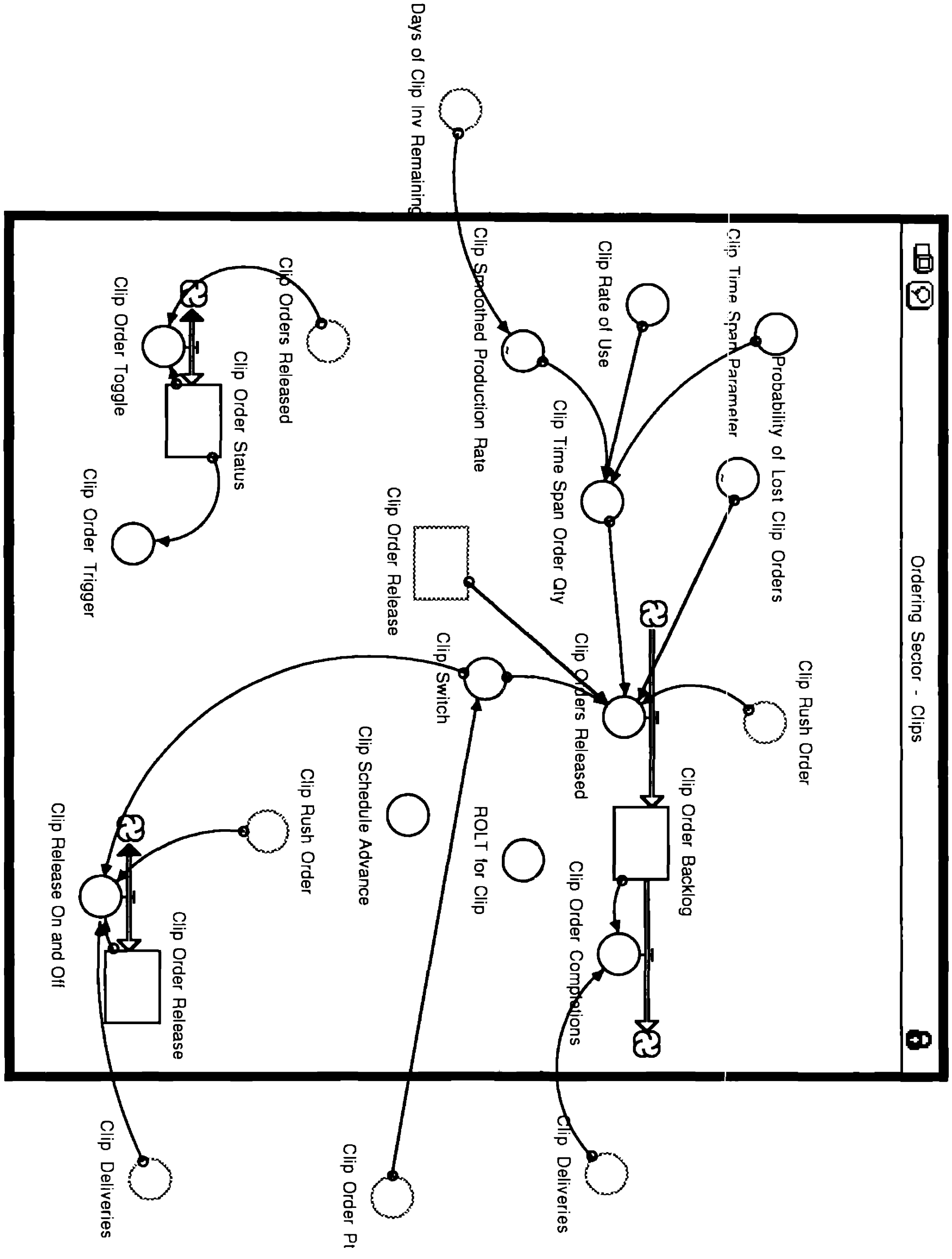

Lot Time - Expediting - Days Until Next Delivery
Days Until Clip Delivery
Clip Cycle Time
Clip Processing Time
Clip Q Ck
Days Until Web Delivery
Web Cycle Time
Web Processing Time
Web Q Ck
Days Until Dbler Delivery
Dbler Cycle Time
Dbler Processing Time
Doubler Q Ck
Days Until Stringer Delivery
Stringer Cycle Time
Stringer Processing Time
Stringer Q Ck

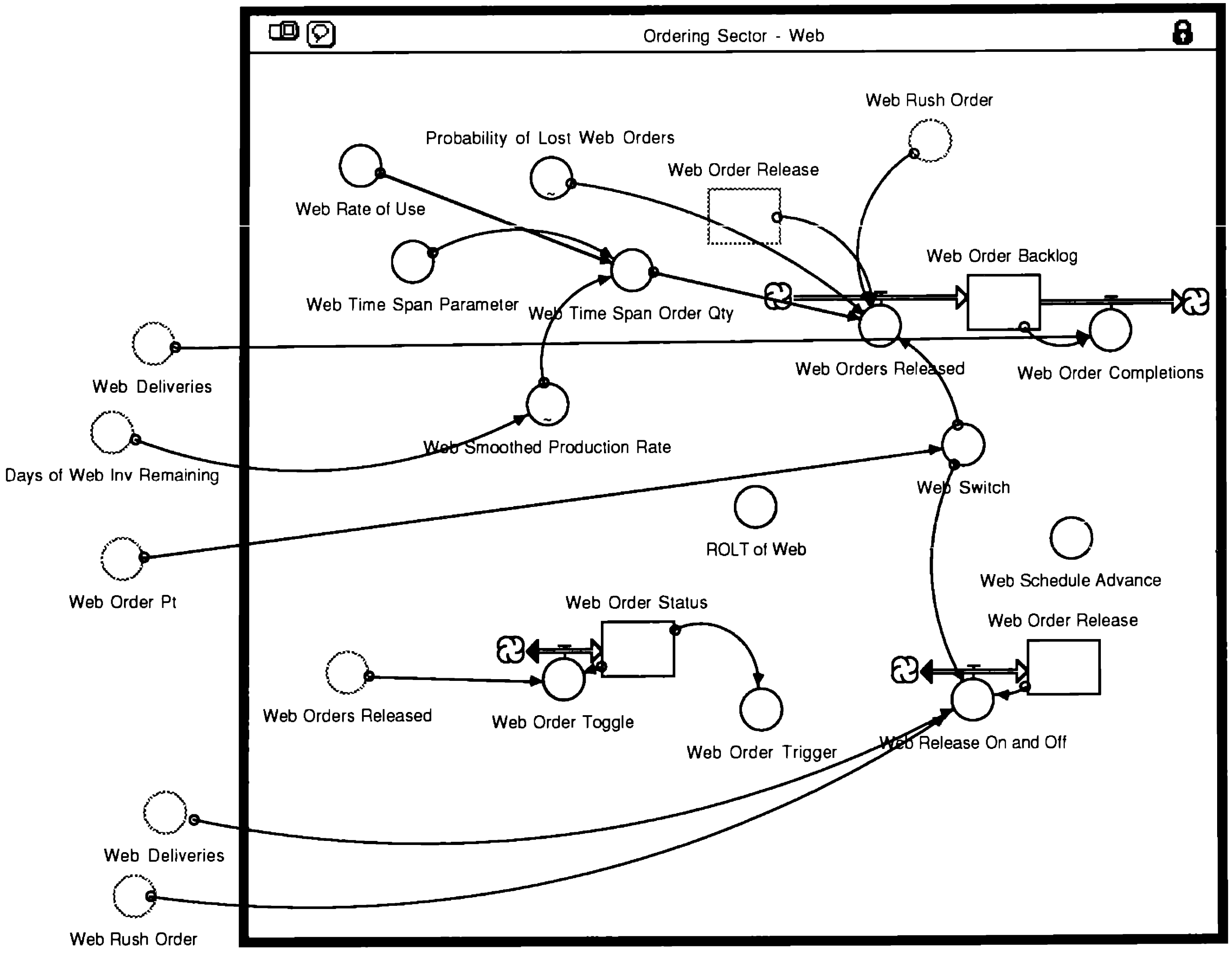

Performance Measurement - Avg Inventory Level
Total SA Inv
Avg SA Inv Level
Inv Flow
PCA 46 Web SA Stores
Total Web Inv
Avg Web Inv Level
Inv Flow 2
PCA 370 Web Stores
Total Clip Inv
Avg Clip Inv Level
Inv Flow 4
PCA 370 Clip Stores
Total Doubler Inv
Avg Doubler Inv Level
Inv Flow 3
PCA 370 Doubler Stores
Total Stringer Inv
Avg Stringer Inv Level
PCA 370 Stringer Stores
Inv Flow 5

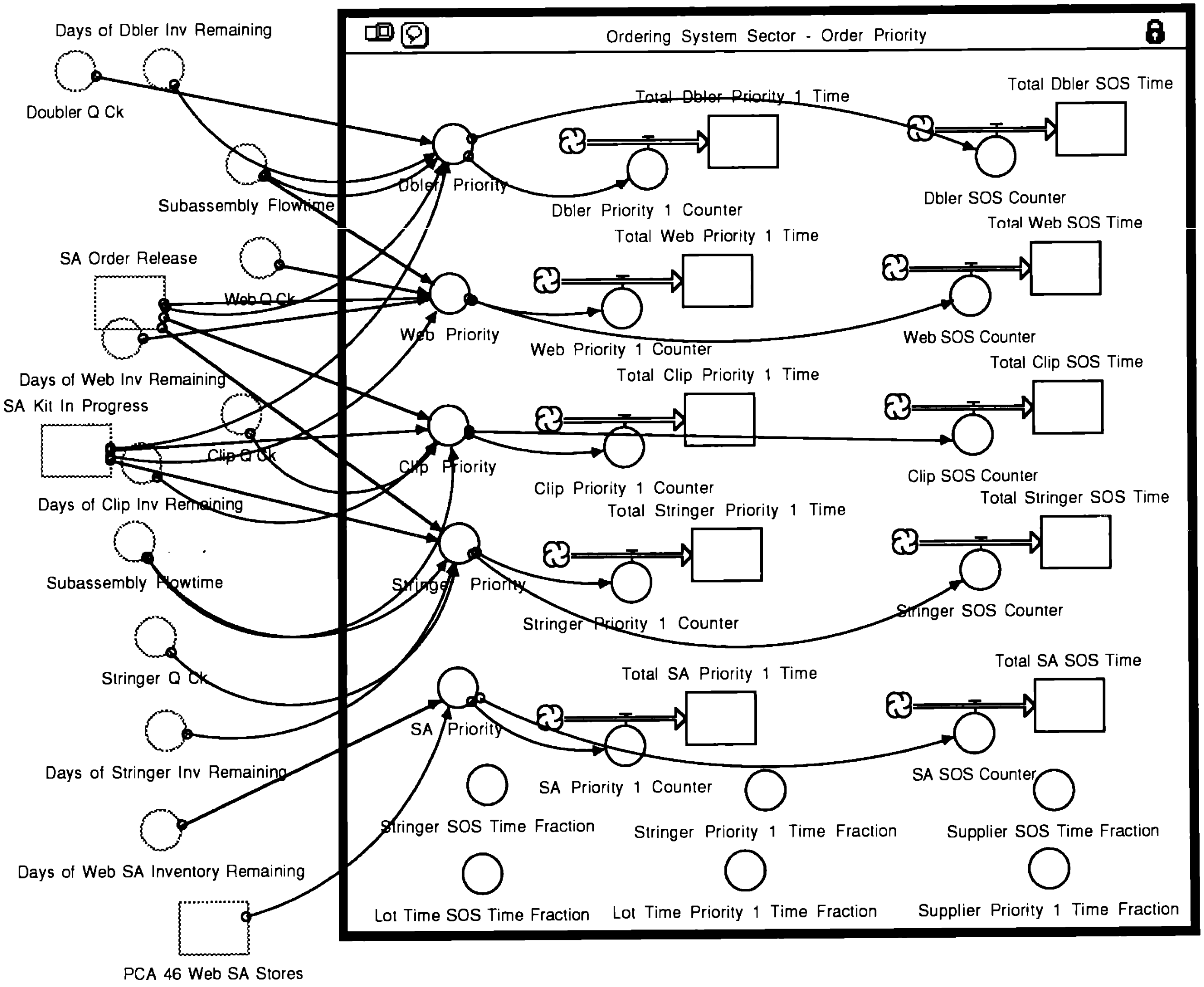

SMC & Suppliers - Expediting
PCA 370 Stringer Stores
Days Until Stringer Delivery
PCA 370 Doubler Stores
Web SA Switch
Days Until Dbler Delivery
Days Until Web Delivery
PCA 370 Web Stores
Web SA Switch
PCA 370 Clip Stores
Days Until Clip Delivery
Stringer Split Order Threshold
Stringer Split Order Signal
Stringer Split Order Reset
Stringer Split Order Qty Fraction
Doubler Split Order Signal
Doubler Split Order Threshold
Doubler Split Order Reset
Doubler Split Order Qtg Fraction
Web Split Order Signal
Web Split Order Threshold
Web Split Order Reset
Web Split Order Qtg Fraction
Clip Split Order Signal
Clip Split Order Threshold
Clip Split Order Reset
Clip Split Order Qtg Fraction
Stringer Split Order Qty
SA Kit In Progress
Doubler Split Order Qty
Split Order Qty Threshold
Web Split Order Qty
SA Kit In Progress
Clip Split Order Qty

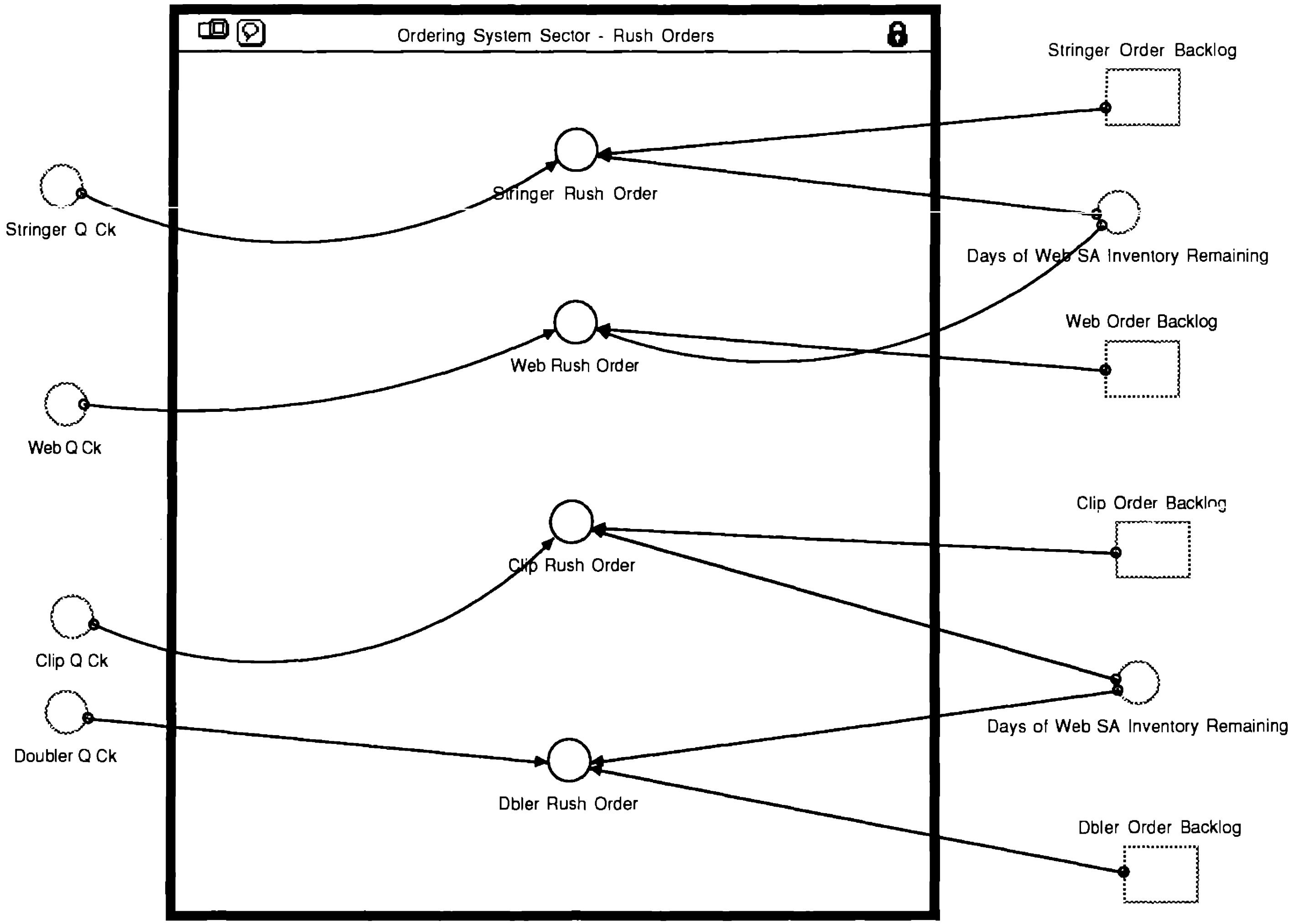

Fabrication - Suppliers - Split Order Doublers & Webs
Total Doubler Split Orders
Doubler Split Order Counter
Doubler Split Order Flowtime
Dbler Split Order Deliveries
Doubler Split Order Qty
Doubler Split Order Flow
Doubler Split Order Delivery
Doubler Split Order Timer
Doubler Split Order Time Flow
Total Web Split Orders
Web Split Order Flowtime
Webs Split Order Counter
Web Split Order Qty
Web Split Order Deliveries
Web Split Order Flow
Web Split Order Delivery
Web Split Order Timer
Web Split Order Time Flow

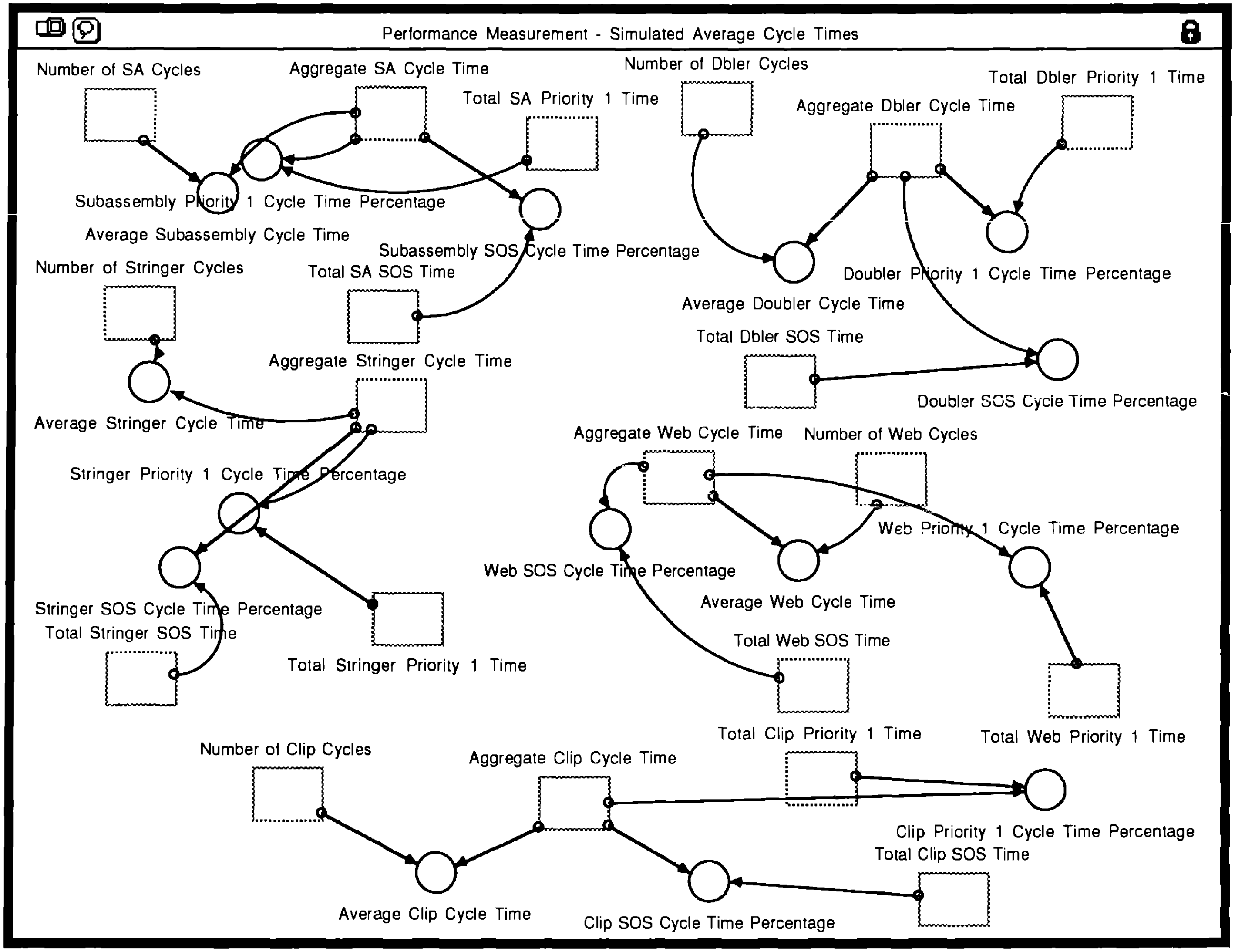

Major Assembly Operations
PCA 46 Web SA Stores
Web SA Arrival Rate
Web SA Usage Rate
Production Rate
SA Q Loss 2
Subassembly Deliveries
Web SA Adjustment Rate
Stockout Trigger
SA Stockout Trigger Activation
SA Inv Loss
Lost SA Inv
Smoothed Production Rate
Total Subassembly Stockouts
Web SA Stockout Flow
Actual Days of SA INV Remaining
PCA 46 Web SA Stores
Monthly Aircraft Production Rate
Production Rate
CC352 Expedite Threshold
Production Rate Test
Historical Production Rate

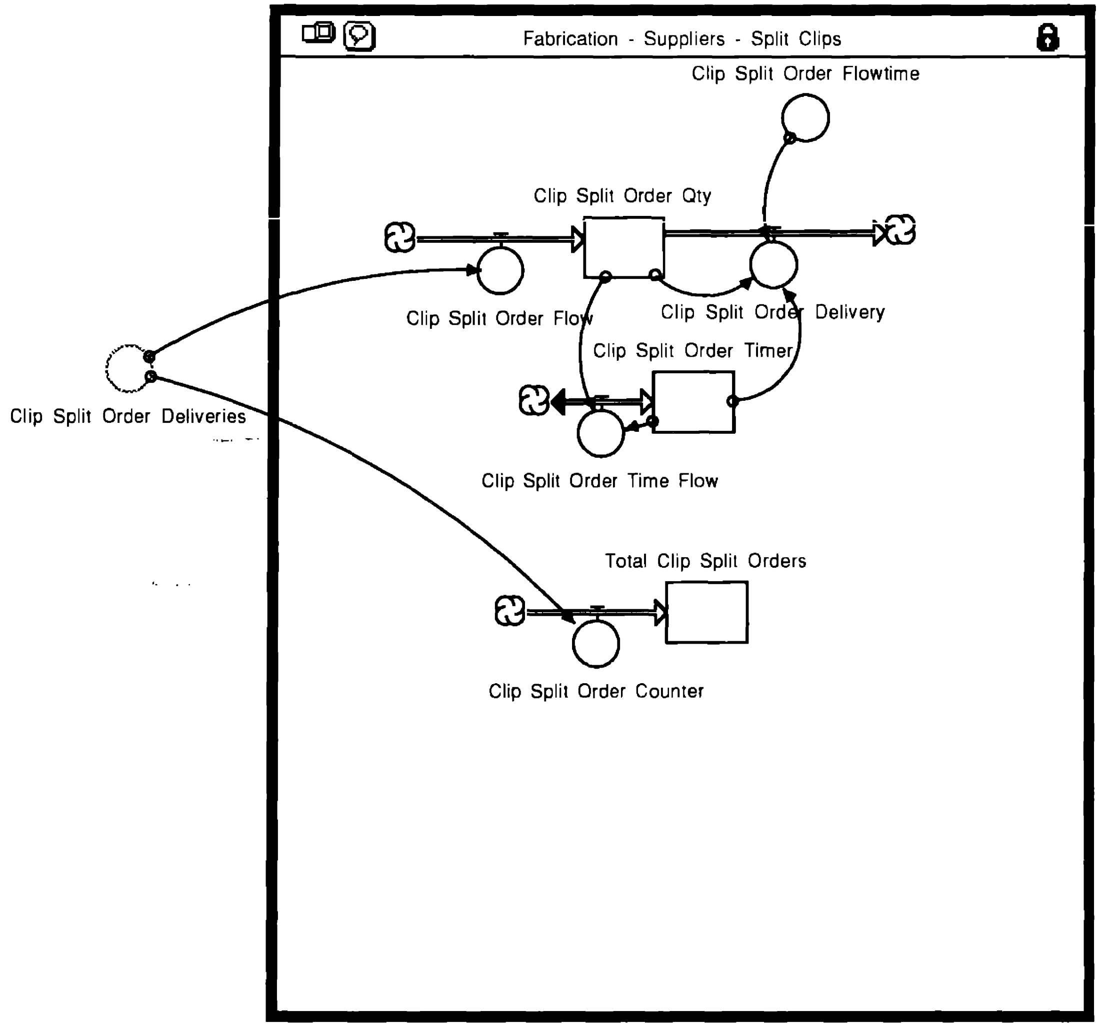

Ordering Sector - Clips
Probability of Lost Clip Orders
Clip Time Span Parameter
Clip Rate of Use
Days of Clip Inv Remaining
Clip Smoothed Production Rate
Clip Time Span Order Qty
Clip Order Release
Clip Orders Released
Clip Rush Order
Clip Switch
Clip Order Backlog
Clip Order Completions
ROLT for Clip
Clip Schedule Advance
Clip Rush Order
Clip Order Release
Clip Release On and Off
Clip Orders Released
Clip Order Status
Clip Order Toggle
Clip Order Trigger
Clip Deliveries
Clip Order Pt
Clip Deliveries


Ordering Sector - Web
Web Rush Order
Probability of Lost Web Orders
Web Order Release
Web Rate of Use
Web Order Backlog
Web Time Span Parameter
Web Time Span Order Qty
Web Deliveries
Web Orders Released
Web Order Completions
Days of Web Inv Remaining
Web Smoothed Production Rate
Web Switch
Web Order Pt
ROLT of Web
Web Schedule Advance
Web Order Status
Web Order Release
Web Orders Released
Web Order Toggle
Web Order Trigger
Web Release On and Off
Web Deliveries
Web Rush Order

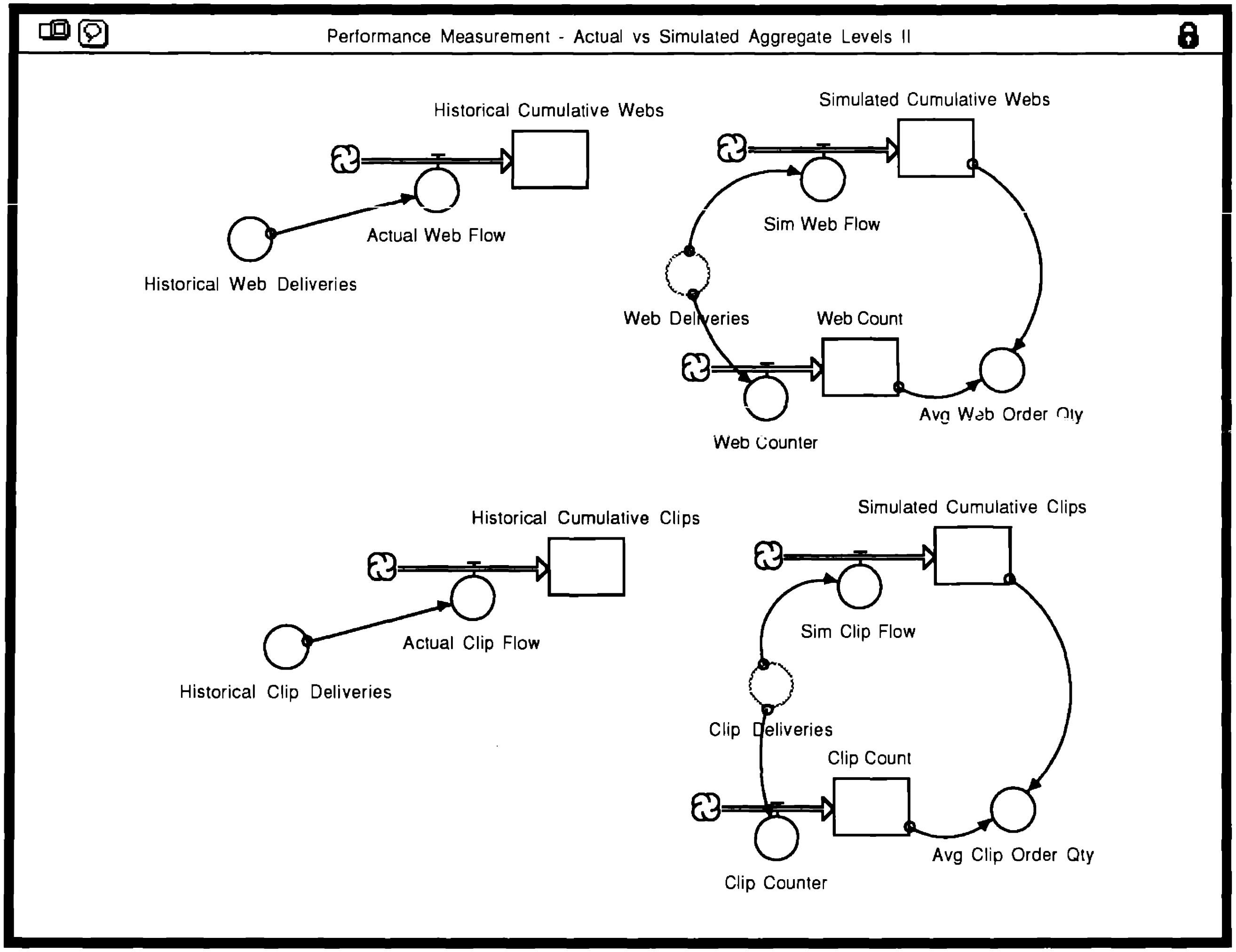

Ordering System Sector - Order Priority
Days of Dbler Inv Remaining
Doubler Q Ck
Subassembly Flowtime
SA Order Release
Web Q Ck
Days of Web Inv Remaining
SA Kit In Progress
Clip Q Ck
Days of Clip Inv Remaining
Subassembly Flowtime
Stringer Q Ck
Days of Stringer Inv Remaining
Days of Web SA Inventory Remaining
PCA 46 Web SA Stores
Dbler Priority
Total Dbler Priority 1 Time
Dbler Priority 1 Counter
Total Dbler SOS Time
Dbler SOS Counter
Web Priority
Total Web Priority 1 Time
Web Priority 1 Counter
Total Web SOS Time
Web SOS Counter
Clip Priority
Total Clip Priority 1 Time
Clip Priority 1 Counter
Total Clip SOS Time
Clip SOS Counter
Stringer Priority
Total Stringer Priority 1 Time
Stringer Priority 1 Counter
Total Stringer SOS Time
Stringer SOS Counter
SA Priority
Total SA Priority 1 Time
SA Priority 1 Counter
Total SA SOS Time
SA SOS Counter
Stringer SOS Time Fraction
Stringer Priority 1 Time Fraction
Supplier SOS Time Fraction
Lot Time SOS Time Fraction
Lot Time Priority 1 Time Fraction
Supplier Priority 1 Time Fraction

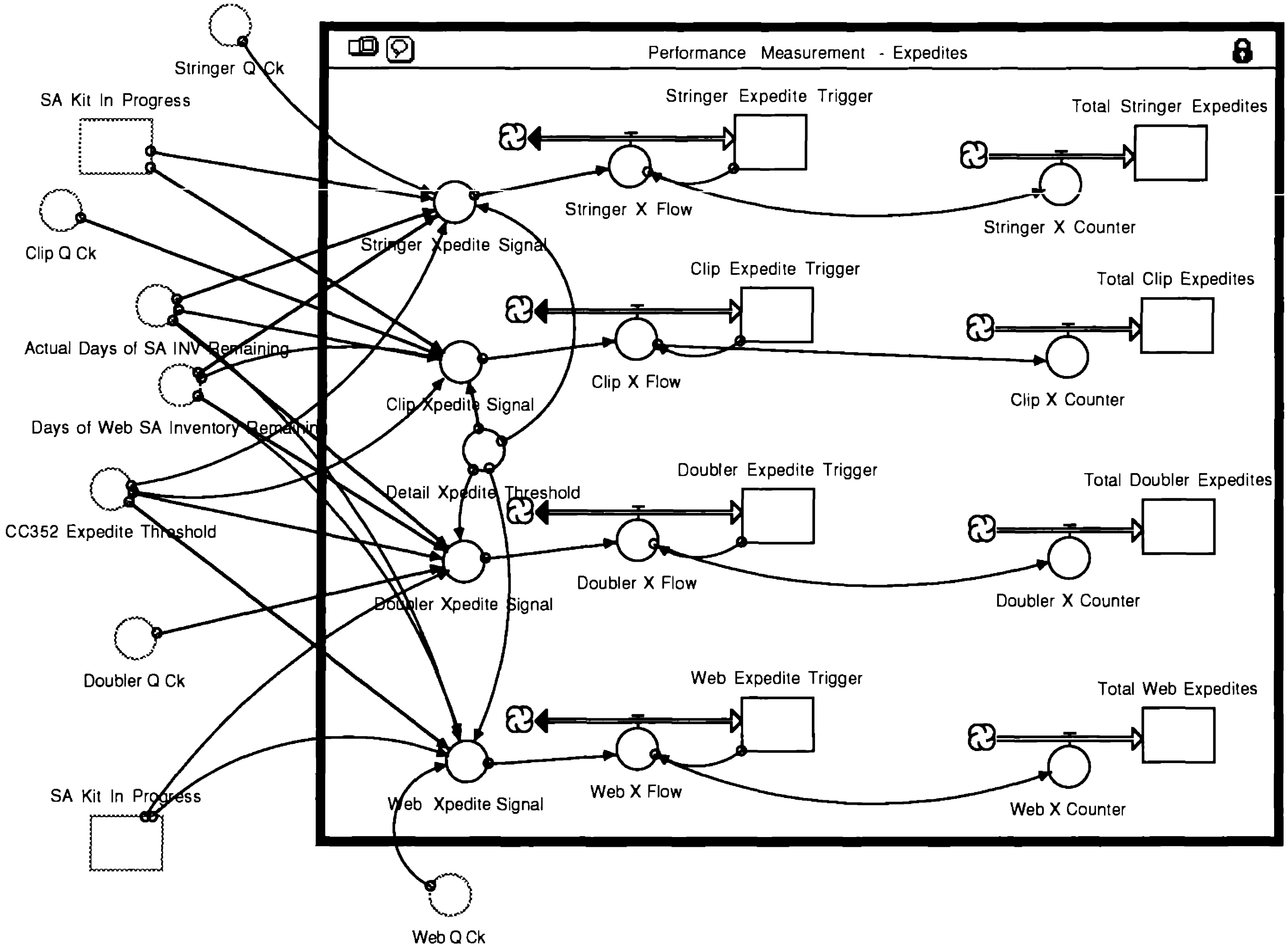

Ordering System Sector - Rush Orders
Stringer Order Backlog
Stringer Rush Order
Stringer Q Ck
Days of Web SA Inventory Remaining
Web Order Backlog
Web Rush Order
Web Q Ck
Clip Order Backlog
Clip Rush Order
Clip Q Ck
Days of Web SA Inventory Remaining
Doubler Q Ck
Dbler Rush Order
Dbler Order Backlog

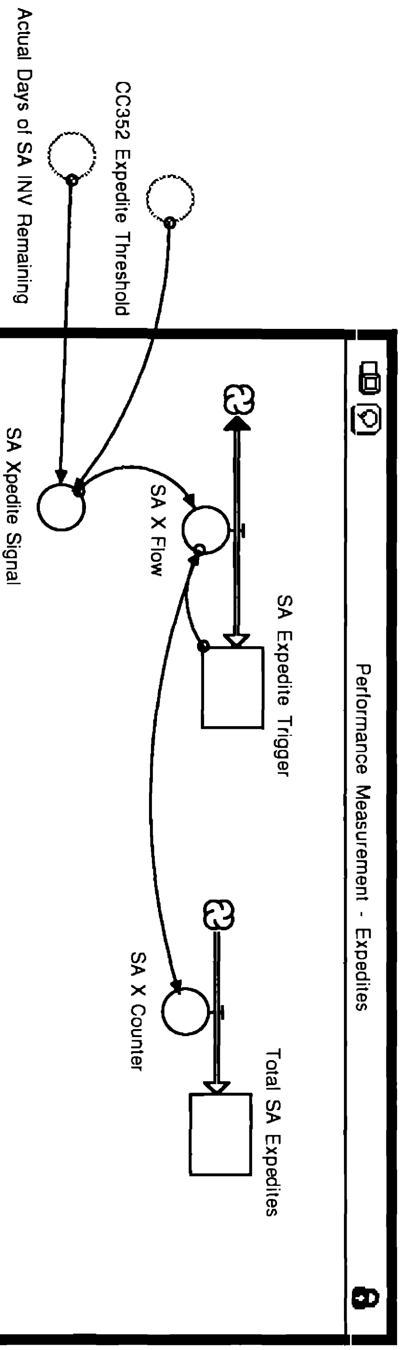

Performance Measurement - Simulated Average Cycle Times
Number of SA Cycles
Aggregate SA Cycle Time
Total SA Priority 1 Time
Subassembly Priority 1 Cycle Time Percentage
Average Subassembly Cycle Time
Subassembly SOS Cycle Time Percentage
Total SA SOS Time
Number of Stringer Cycles
Aggregate Stringer Cycle Time
Average Stringer Cycle Time
Stringer Priority 1 Cycle Time Percentage
Stringer SOS Cycle Time Percentage
Total Stringer SOS Time
Total Stringer Priority 1 Time
Number of Dbler Cycles
Total Dbler Priority 1 Time
Aggregate Dbler Cycle Time
Doubler Priority 1 Cycle Time Percentage
Average Doubler Cycle Time
Total Dbler SOS Time
Doubler SOS Cycle Time Percentage
Aggregate Web Cycle Time
Number of Web Cycles
Web Priority 1 Cycle Time Percentage
Web SOS Cycle Time Percentage
Average Web Cycle Time
Total Web SOS Time
Total Clip Priority 1 Time
Total Web Priority 1 Time
Number of Clip Cycles
Aggregate Clip Cycle Time
Clip Priority 1 Cycle Time Percentage
Total Clip SOS Time
Average Clip Cycle Time
Clip SOS Cycle Time Percentage

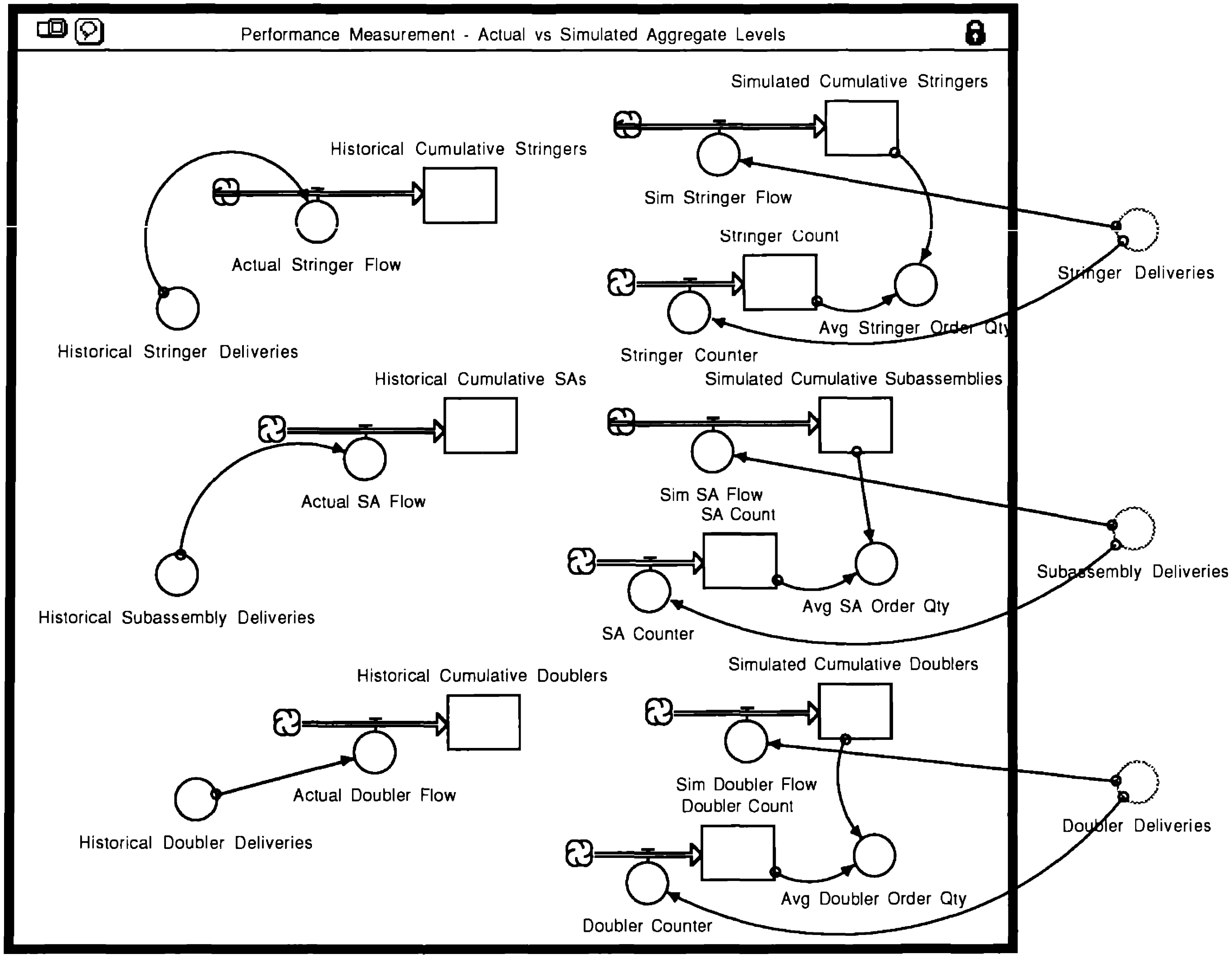


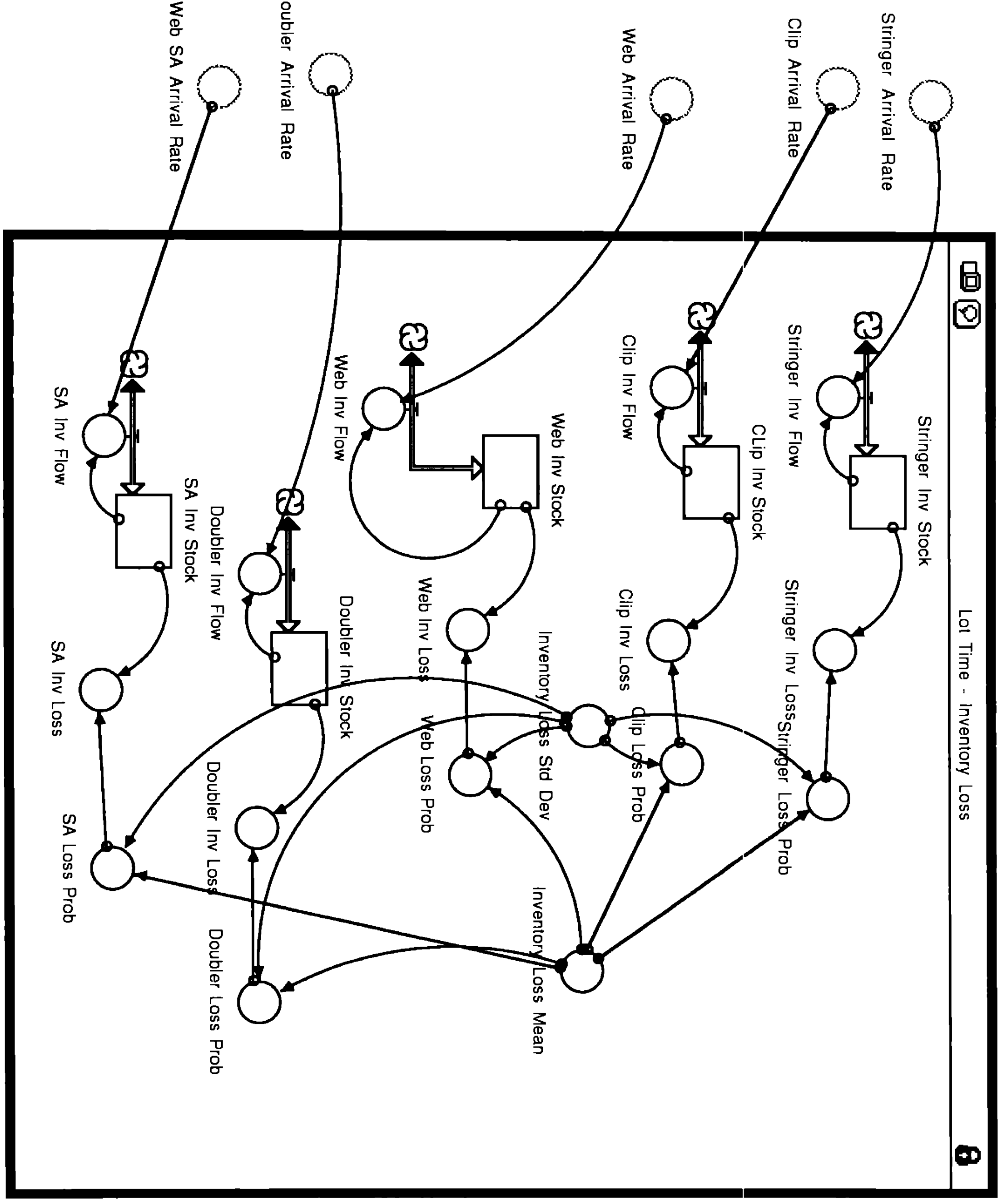
Fabrication - Suppliers - Split Clips
Clip Split Order Flowtime
Clip Split Order Qty
Clip Split Order Flow
Clip Split Order Delivery
Clip Split Order Timer
Clip Split Order Deliveries
Clip Split Order Time Flow
Total Clip Split Orders
Clip Split Order Counter

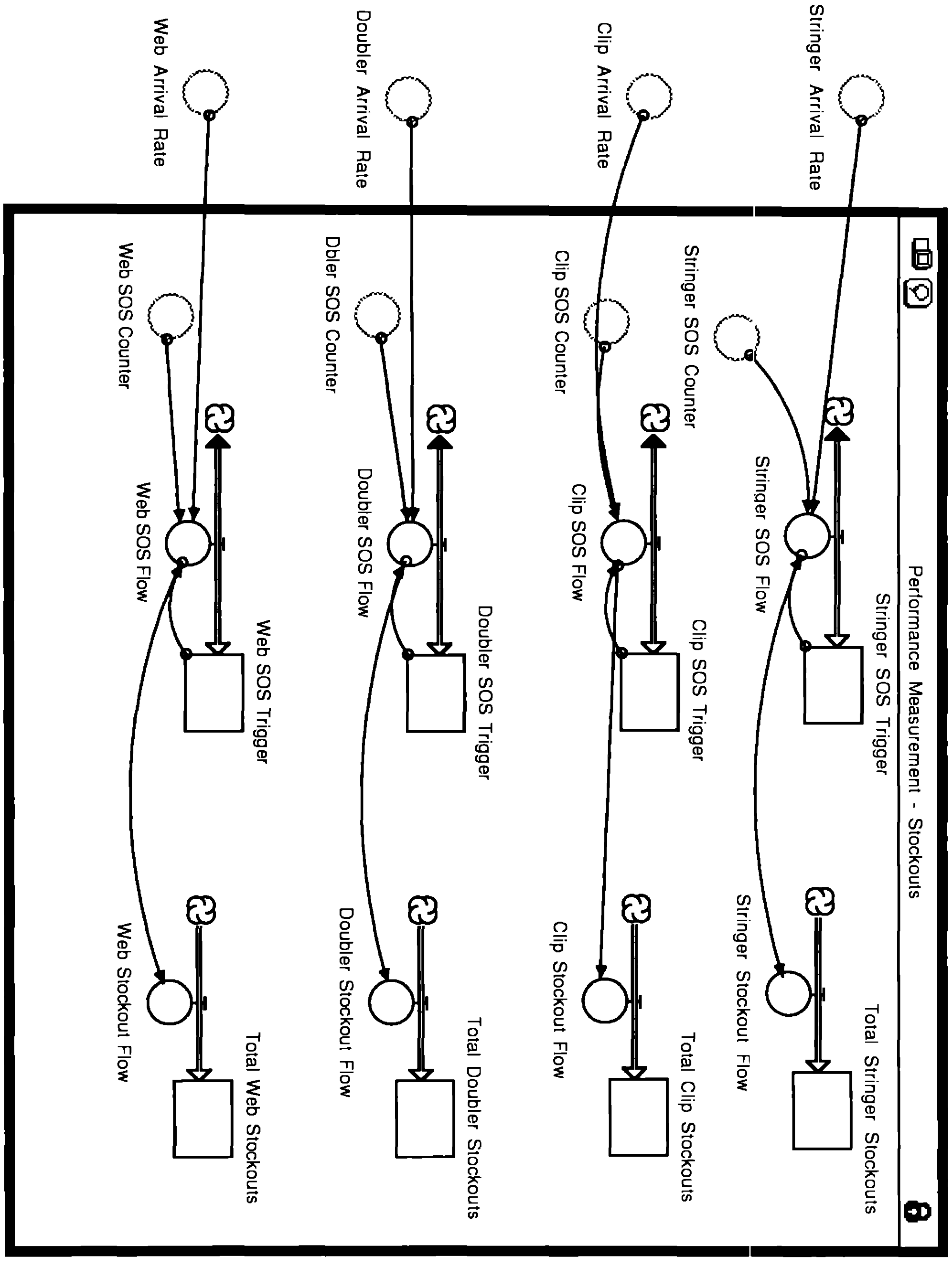

Performance Measurement - Actual vs Simulated Aggregate Levels II
Historical Cumulative Webs
Actual Web Flow
Historical Web Deliveries
Simulated Cumulative Webs
Sim Web Flow
Web Deliveries
Web Count
Avg Web Order Qty
Web Counter
Historical Cumulative Clips
Actual Clip Flow
Historical Clip Deliveries
Simulated Cumulative Clips
Sim Clip Flow
Clip Deliveries
Clip Count
Avg Clip Order Qty
Clip Counter

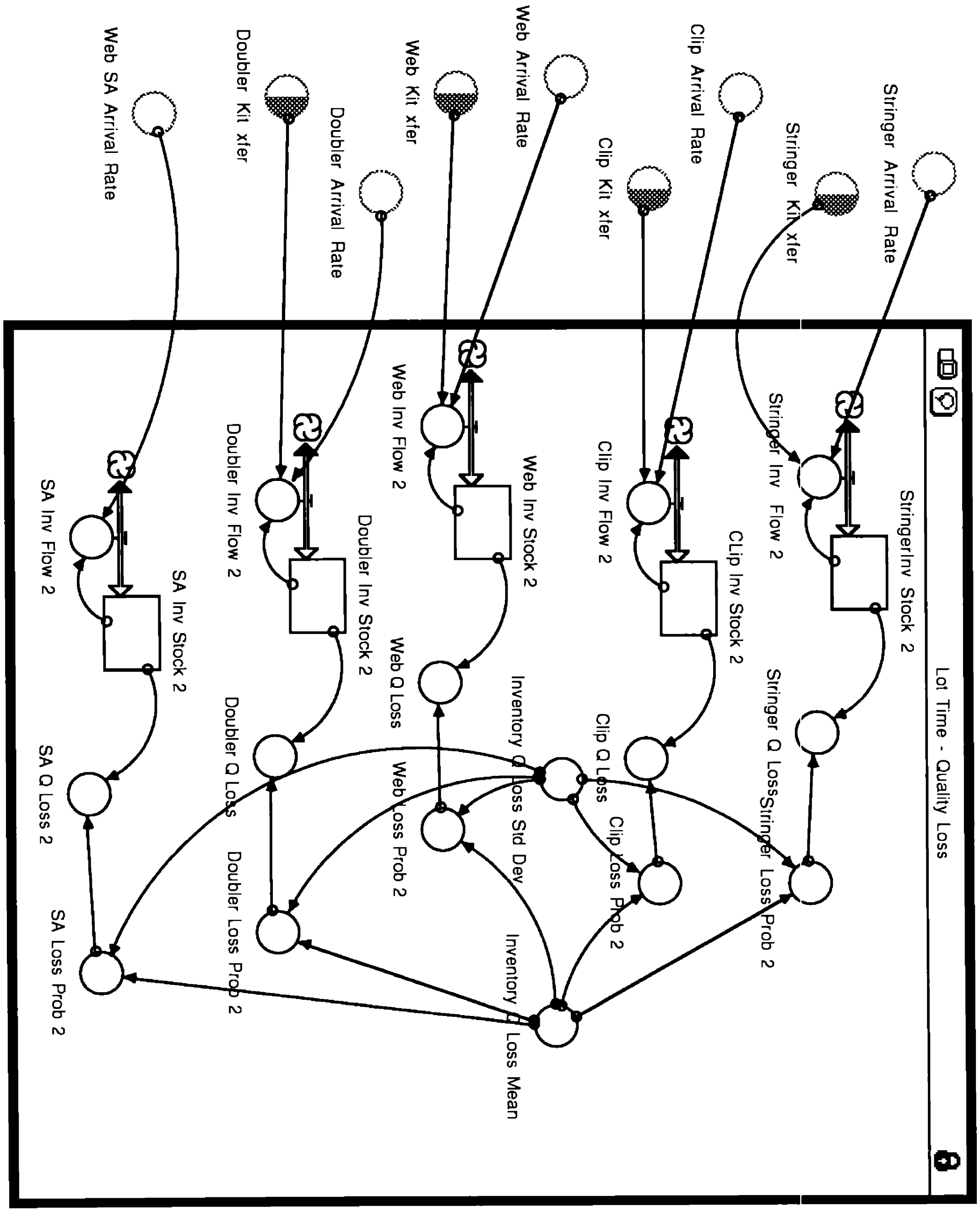

Performance Measurement - Expedites
Stringer Q Ck
SA Kit In Progress
Clip Q Ck
Actual Days of SA INV Remaining
Days of Web SA Inventory Remaining
CC352 Expedite Threshold
Doubler Q Ck
SA Kit In Progress
Web Q Ck
Stringer Xpedite Signal
Clip Xpedite Signal
Detail Xpedite Threshold
Doubler Xpedite Signal
Web Xpedite Signal
Stringer Expedite Trigger
Stringer X Flow
Clip Expedite Trigger
Clip X Flow
Doubler Expedite Trigger
Doubler X Flow
Web Expedite Trigger
Web X Flow
Total Stringer Expedites
Stringer X Counter
Total Clip Expedites
Clip X Counter
Total Doubler Expedites
Doubler X Counter
Total Web Expedites
Web X Counter


Performance Measurement - Expedites
SA Expedite Trigger
Total SA Expedites
SA X Flow
SA X Counter
SA Xpedite Signal
CC352 Expedite Threshold
Actual Days of SA INV Remaining


Performance Measurement - Actual vs Simulated Aggregate Levels
Historical Cumulative Stringers
Actual Stringer Flow
Historical Stringer Deliveries
Simulated Cumulative Stringers
Sim Stringer Flow
Stringer Count
Avg Stringer Order Qty
Stringer Counter
Stringer Deliveries
Historical Cumulative SAs
Actual SA Flow
Historical Subassembly Deliveries
Simulated Cumulative Subassemblies
Sim SA Flow
SA Count
Avg SA Order Qty
SA Counter
Subassembly Deliveries
Historical Cumulative Doublers
Actual Doubler Flow
Historical Doubler Deliveries
Simulated Cumulative Doublers
Sim Doubler Flow
Doubler Count
Avg Doubler Order Qty
Doubler Counter
Doubler Deliveries


Lot Time - Inventory Loss
Stringer Arrival Rate
Clip Arrival Rate
Web Arrival Rate
Doubler Arrival Rate
Web SA Arrival Rate
Stringer Inv Flow
Stringer Inv Stock
Stringer Inv Loss
Stringer Loss Prob
Clip Inv Flow
CLip Inv Stock
Clip Inv Loss
Clip Loss Prob
Web Inv Flow
Web Inv Stock
Web Inv Loss
Web Loss Prob
Inventory Loss Std Dev
Inventory Loss Mean
Doubler Inv Flow
Doubler Inv Stock
Doubler Inv Loss
Doubler Loss Prob
SA Inv Flow
SA Inv Stock
SA Inv Loss
SA Loss Prob


Performance Measurement - Stockouts
Stringer Arrival Rate
Clip Arrival Rate
Doubler Arrival Rate
Web Arrival Rate
Stringer SOS Counter
Clip SOS Counter
Dbler SOS Counter
Web SOS Counter
Stringer SOS Flow
Clip SOS Flow
Doubler SOS Flow
Web SOS Flow
Stringer SOS Trigger
Clip SOS Trigger
Doubler SOS Trigger
Web SOS Trigger
Stringer Stockout Flow
Clip Stockout Flow
Doubler Stockout Flow
Web Stockout Flow
Total Stringer Stockouts
Total Clip Stockouts
Total Doubler Stockouts
Total Web Stockouts


Lot Time - Quality Loss
Stringer Arrival Rate
Stringer Kit xfer
Clip Arrival Rate
Clip Kit xfer
Web Arrival Rate
Web Kit xfer
Doubler Arrival Rate
Doubler Kit xfer
Web SA Arrival Rate
StringerInv Stock 2
Stringer Inv Flow 2
CLip Inv Stock 2
Clip Inv Flow 2
Web Inv Stock 2
Web Inv Flow 2
Doubler Inv Stock 2
Doubler Inv Flow 2
SA Inv Stock 2
SA Inv Flow 2
Stringer Q Loss
Clip Q Loss
Web Q Loss
Inventory Q Loss Std Dev
Doubler Q Loss
SA Q Loss 2
Stringer Loss Prob 2
Clip Loss Prob 2
Web Loss Prob 2
Doubler Loss Prob 2
SA Loss Prob 2
Inventory Q Loss Mean

Appendix A3 - Model Equations & Documentation

```
Lot Time
Aggregate_SA_Cycle_Time(t) = Aggregate_SA_Cycle_Time(t - dt) +
(Aggregate_SA_Cycle_Time_Flow) * dt
INIT Aggregate_SA_Cycle_Time = .001

INFLOWS:
Aggregate_SA_Cycle_Time_Flow = if (Subassembly_Deliveries > 0) then
 Subassembly_Cycle_Time
else
 0
INFLOW TO:
Kits_Awaiting_Assembly(t) = Kits_Awaiting_Assembly(t - dt) +
(xfer_to_Work_Area - Subassembly_Deliveries) * dt
INIT Kits_Awaiting_Assembly = 0

INFLOWS:
xfer_to_Work_Area = CONVEYOR OUTFLOW
OUTFLOW FROM:
INFLOW TO:
OUTFLOWS:
Subassembly_Deliveries = if (Subassembly_Cycle_Time >=
Lot_Time_Processing_Time)  then
 Kits_Awaiting_Assembly
else
 0
OUTFLOW FROM:
Kits_in_Storage(t) = Kits_in_Storage(t - dt) + (Stringer_Kit_xfer +
Web_Kit_xfer + Clip_Kit_xfer + Doubler_Kit_xfer - xfer_to_Work_Area) *
dt
INIT Kits_in_Storage = 0
        TRANSIT TIME = 1
        ENTRANCE CAPACITY = INF
INFLOWS:
Stringer_Kit_xfer(i) = Stringer_Kit_xfer(o) * CONVERSION MULTIPLIER
        CONVERSION MULTIPLIER = 0.25

OUTFLOW FROM:
INFLOW TO:
Web_Kit_xfer(i) = Web_Kit_xfer(o) * CONVERSION MULTIPLIER
        CONVERSION MULTIPLIER = 0.25

OUTFLOW FROM:
INFLOW TO:
```

```
Clip_Kit_xfer(i) = Clip_Kit_xfer(o) * CONVERSION MULTIPLIER
        CONVERSION MULTIPLIER = 0.25

OUTFLOW FROM:
INFLOW TO:
Doubler_Kit_xfer(i) = Doubler_Kit_xfer(o) * CONVERSION MULTIPLIER
        CONVERSION MULTIPLIER = 0.25

OUTFLOW FROM:
INFLOW TO:
OUTFLOWS:
xfer_to_Work_Area = CONVEYOR OUTFLOW
OUTFLOW FROM:
INFLOW TO:
Number_of_SA_Cycles(t) = Number_of_SA_Cycles(t - dt) +
(Flow_of_SA_Cycles) * dt
INIT Number_of_SA_Cycles = .001

INFLOWS:
Flow_of_SA_Cycles = if (Subassembly_Deliveries > 0) then
 1
else
 0
INFLOW TO:
PCA_370_Clip_Stores(t) = PCA_370_Clip_Stores(t - dt) + (Clip_Arrival_Rate
- Clip_Kit_xfer - Clip_Adjustment_Rate - Lost_Clip_Inv) * dt
INIT PCA_370_Clip_Stores =
round((Clip_Time_Span_Parameter*Smoothed_Production_Rate)+.5)

INFLOWS:
Clip_Arrival_Rate = if (Clip_Deliveries > 0) then
 Clip_Deliveries
else
if (Clip_Split_Order_Delivery >0) then
 Clip_Split_Order_Delivery
else
 0
INFLOW TO:
OUTFLOWS:
Clip_Kit_xfer(o) = if (SA_Kit_Release =1) and (SA_Split_Order_Kit_Release
= 1) then
 Min(Kit_Split_Order_Quantity,SA_Order_Backlog)
else
if (SA_Kit_Release =1) then
SA_Order_Backlog
else
```

```
0
OUTFLOW FROM:
INFLOW TO:
Clip_Adjustment_Rate = if (Clip_Kit_xfer > 0) then
 min(PCA_370_Clip_Stores,Clip_Q_Loss)
else
 0
OUTFLOW FROM:
Lost_Clip_Inv = Clip_Inv_Loss
OUTFLOW FROM:
PCA_370_Doubler_Stores(t) = PCA_370_Doubler_Stores(t - dt) +
(Doubler_Arrival_Rate - Doubler_Kit_xfer - Doubler_Adjustment_Rate -
Lost_Dbler_Inv) * dt
INIT PCA_370_Doubler_Stores =
Round((Doubler_Time_Span_Parameter*Smoothed_Production_Rate)+.5)

INFLOWS:
Doubler_Arrival_Rate = if (Doubler_Deliveries > 0) then
 Doubler_Deliveries
else
if (Doubler_Split_Order_Delivery > 0) then
 Doubler_Split_Order_Delivery
else
 0
INFLOW TO:
OUTFLOWS:
Doubler_Kit_xfer(o) = if (SA_Kit_Release =1) and
(SA_Split_Order_Kit_Release = 1) then
 Min(Kit_Split_Order_Quantity,SA_Order_Backlog)
else
if (SA_Kit_Release =1) then
SA_Order_Backlog
else
0
OUTFLOW FROM:
INFLOW TO:
Doubler_Adjustment_Rate = if  (Doubler_Kit_xfer > 0)  then
 min(PCA_370_Doubler_Stores,Doubler_Q_Loss)
else
 0
OUTFLOW FROM:
Lost_Dbler_Inv = Doubler_Inv_Loss
OUTFLOW FROM:
PCA_370_Stringer_Stores(t) = PCA_370_Stringer_Stores(t - dt) +
(Stringer_Arrival_Rate - Stringer_Kit_xfer - Stringer_Adjustment_Rate -
Lost_Stringer_Inv) * dt
```

```
INIT PCA_370_Stringer_Stores =
Round((Stringer_Time_Span_Parameter*Smoothed_Production_Rate)+.5)

INFLOWS:
Stringer_Arrival_Rate = if (Stringer_Deliveries > 0) then
 Stringer_Deliveries
else
 if (Stringer_Split_Order_Delivery > 0) then
 Stringer_Split_Order_Delivery
else
 0
INFLOW TO:
OUTFLOWS:
Stringer_Kit_xfer(o) = if (SA_Kit_Release =1) and
(SA_Split_Order_Kit_Release = 1) then
 Min(Kit_Split_Order_Quantity,SA_Order_Backlog)
else
if (SA_Kit_Release = 1)then
 SA_Order_Backlog
else
0
OUTFLOW FROM:
INFLOW TO:
Stringer_Adjustment_Rate = if (Stringer_Kit_xfer > 0) then
 min(PCA_370_Stringer_Stores,Stringer_Q_Loss)
else
 0
OUTFLOW FROM:
Lost_Stringer_Inv = Stringer_Inv_Loss
OUTFLOW FROM:
PCA_370_Web_Stores(t) = PCA_370_Web_Stores(t - dt) +
(Web_Arrival_Rate - Web_Kit_xfer - Web_Adjustment_Rate -
Lost_Web_Inv) * dt
INIT PCA_370_Web_Stores =
round((Web_Time_Span_Parameter*Smoothed_Production_Rate)+.5)

INFLOWS:
Web_Arrival_Rate = if (Web_Deliveries > 0) then
 Web_Deliveries
else
if (Web_Split_Order_Delivery > 0) then
 Web_Split_Order_Delivery
else
 0
INFLOW TO:
OUTFLOWS:
```

```
Web_Kit_xfer(o) = if (SA_Kit_Release =1) and (SA_Split_Order_Kit_Release
= 1) then
 Min(Kit_Split_Order_Quantity,SA_Order_Backlog)
else
if (SA_Kit_Release =1) then
SA_Order_Backlog
else
0
OUTFLOW FROM:
INFLOW TO:
Web_Adjustment_Rate = if (Web_Kit_xfer > 0)  then
 min(PCA_370_Web_Stores,Web_Q_Loss)
else
 0
OUTFLOW FROM:
Lost_Web_Inv = Web_Inv_Loss
OUTFLOW FROM:
Subassembly_Cycle_Time(t) = Subassembly_Cycle_Time(t - dt) +
(Processing_Counter) * dt
INIT Subassembly_Cycle_Time = 0

INFLOWS:
Processing_Counter = if (Kits_Awaiting_Assembly > 0) then
 1
else
 -Subassembly_Cycle_Time
INFLOW TO:
Lot_Time_Processing_Time = if (SA_Priority = 0) then
 Lot_Time_SOS_Time_Fraction *
Normal(Mean_of_Processing,Std_Dev_of_Processing,111)
else
if (SA_Priority = 1) then
 Lot_Time_Priority_1_Time_Fraction *
Normal(Mean_of_Processing,Std_Dev_of_Processing,111)
else
 Normal(Mean_of_Processing,Std_Dev_of_Processing,111)
Mean_of_Processing = 1
Std_Dev_of_Processing = .25

Major Assembly Operations
PCA_46_Web_SA_Stores(t) = PCA_46_Web_SA_Stores(t - dt) +
(Web_SA_Arrival_Rate - Web_SA_Usage_Rate -
Web_SA_Adjustment_Rate - Lost_SA_Inv) * dt
INIT PCA_46_Web_SA_Stores =
round((Smoothed_Production_Rate*SA_Time_Span_Parameter)+.5)
```

```
INFLOWS:
Web_SA_Arrival_Rate = if (PCA_46_Web_SA_Stores < 0) then
 Subassembly_Deliveries + PCA_46_Web_SA_Stores
else
 Subassembly_Deliveries
INFLOW TO:
OUTFLOWS:
Web_SA_Usage_Rate = Production_Rate
OUTFLOW FROM:
Web_SA_Adjustment_Rate =
min(SA_Q_Loss_2,PCA_46_Web_SA_Stores)
OUTFLOW FROM:
Lost_SA_Inv = min(SA_Inv_Loss,PCA_46_Web_SA_Stores)
OUTFLOW FROM:
Stockout_Trigger(t) = Stockout_Trigger(t - dt) +
(SA_Stockout_Trigger_Activation) * dt
INIT Stockout_Trigger = 0

INFLOWS:
SA_Stockout_Trigger_Activation = if (PCA_46_Web_SA_Stores <= 0) and
(Stockout_Trigger = 0) then
 1
else
if (PCA_46_Web_SA_Stores > 0) then
 -Stockout_Trigger
else
 0
INFLOW TO:
Total_Subassembly_Stockouts(t) = Total_Subassembly_Stockouts(t - dt) +
(Web_SA_Stockout_Flow) * dt
INIT Total_Subassembly_Stockouts = 0

INFLOWS:
Web_SA_Stockout_Flow = if (SA_Stockout_Trigger_Activation = 1) then
 1
else
 0
INFLOW TO:
Actual_Days_of_SA_INV_Remaining =
round(PCA_46_Web_SA_Stores/Smoothed_Production_Rate)
CC352_Expedite_Threshold = 5
Historical_Production_Rate = step(.25,0) + step(-.25,24) + step(.25,28) + step(-
.25,56) + step(.25,66) + step(.08,106) + step(-.08,109) + step(-.25,113) +
step(.25,147) + step(-.25,151) + step(.25,158) + step(-.25,174) - step(1,177) + step(-
.5,178) + step(-.17,180) + step(-.33,285) + step(.33,288) + step(-.33,327) +
step(.33,330) + step(-.33,339) + step(.33,342) + step(-.33,351) + step(.33,354) +
```

```
step(-.33,363) + step(.33,366) + step(-.33,375) + step(.33,378) + step(-.33,387) +
step(.33,390) + step(-.33,405) + step(.33,408) + step(-.08,447) + step(.08,451) +
step(-.33,571) + step(.33,574) + step(-.33,583) + step(.33,586) + step(-.33,595) +
step(.33,598) + step(-.33,607) + step(.33,610) + step(-.33,619) + step(.33,622) +
step(.07,628) + step(-.4,652) + step(.4,661) + step(-.4,796) + step(.4,798) + step(-
.4,806) + step(.4,809) + step(-.4,816) + step(.4,818) + step(-.4,826) + step(.4,829) +
step(-.4,836) + step(.4,838)  + step(-.4,904) + step(.4,913) + step(-.07,983)
Monthly_Aircraft_Production_Rate = Production_Rate*21
Production_Rate = Historical_Production_Rate
Production_Rate_Test = .33
Smoothed_Production_Rate = GRAPH(Time)
(0.00, 0.25), (9.90, 0.25), ...

Fabrication - Stringer
Aggregate_Stringer_Cycle_Time(t) = Aggregate_Stringer_Cycle_Time(t - dt)
+ (Aggregate_Stringer_Cycle_Time_Flow) * dt
INIT Aggregate_Stringer_Cycle_Time = .001

INFLOWS:
Aggregate_Stringer_Cycle_Time_Flow = if (Stringer_Deliveries > 0) then
 Stringer_Cycle_Time
else
 0
INFLOW TO:
Number_of_Stringer_Cycles(t) = Number_of_Stringer_Cycles(t - dt) +
(Flow_of_Stringer_Cycles) * dt
INIT Number_of_Stringer_Cycles = .001

INFLOWS:
Flow_of_Stringer_Cycles = if (Stringer_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Stringers_Being_Fabricated(t) = Stringers_Being_Fabricated(t - dt) +
(Stringer_Incoming_Orders - Stringer_Deliveries -
Stringer_Split_Order_Deliveries) * dt
INIT Stringers_Being_Fabricated = 0

INFLOWS:
Stringer_Incoming_Orders = if (Stringer_Order_Trigger=1) then
 Stringer_Order_Backlog
else
 0
INFLOW TO:
OUTFLOWS:
```

```
Stringer_Deliveries = if (Stringer_Cycle_Time >= Stringer_Processing_Time)
then
 Stringers_Being_Fabricated
else
 0
OUTFLOW FROM:
Stringer_Split_Order_Deliveries = if (Stringer_Split_Order_Signal = 1) and
(Stringer_Split_Order_Qty = 0) then
 Round(Stringer_Split_Order_Qty_Fraction * Stringers_Being_Fabricated)
else
 0

OUTFLOW FROM:
Stringer_Cycle_Time(t) = Stringer_Cycle_Time(t - dt) +
(Stringer_Time_Increment) * dt
INIT Stringer_Cycle_Time = 0

INFLOWS:
Stringer_Time_Increment = if (Stringers_Being_Fabricated>0) then
 1
else
 -Stringer_Cycle_Time

INFLOW TO:
Stringer_Avg_Processing_Time = 5
Stringer_Order_Time_Adjust = if (Stringer_Priority=0) then
 Stringer_SOS_Time_Fraction
else
 if (Stringer_Priority=1) then
  Stringer_Priority_1_Time_Fraction
 else
  1

Stringer_Processing_Time =
Normal(Stringer_Avg_Processing_Time,Stringer_Std_Dev_Processing_Ti
me,222)*Stringer_Order_Time_Adjust
Stringer_Std_Dev_Processing_Time = 1

Fabrication - Doubler
Aggregate_Dbler_Cycle_Time(t) = Aggregate_Dbler_Cycle_Time(t - dt) +
(Aggregate_Dbler_Cycle_Time_Flow) * dt
INIT Aggregate_Dbler_Cycle_Time = .001

INFLOWS:
Aggregate_Dbler_Cycle_Time_Flow = if (Doubler_Deliveries > 0) then
 Dbler_Cycle_Time
```

```
else
 0
INFLOW TO:
Dblers_in_Fabrication(t) = Dblers_in_Fabrication(t - dt) +
(Dbler_Incoming_Orders - Doubler_Deliveries -
Dbler_Split_Order_Deliveries) * dt
INIT Dblers_in_Fabrication = 0

INFLOWS:
Dbler_Incoming_Orders = if (Dbler_Order_Trigger=1) then
 Dbler_Order_Backlog
else
 0
INFLOW TO:
OUTFLOWS:
Doubler_Deliveries = if (Dbler_Cycle_Time >= Dbler_Processing_Time) then
 Dblers_in_Fabrication
else
 0
OUTFLOW FROM:
Dbler_Split_Order_Deliveries = if (Doubler_Split_Order_Signal = 1) and
(Doubler_Split_Order_Qty = 0) then
 Round(Doubler_Split_Order_Qtg_Fraction * Dblers_in_Fabrication)
else
 0

OUTFLOW FROM:
Dbler_Cycle_Time(t) = Dbler_Cycle_Time(t - dt) + (Dbler_Time_Increment)
* dt
INIT Dbler_Cycle_Time = 0

INFLOWS:
Dbler_Time_Increment = if (Dblers_in_Fabrication>0) then
 1
else
 -Dbler_Cycle_Time

INFLOW TO:
Number_of_Dbler_Cycles(t) = Number_of_Dbler_Cycles(t - dt) +
(Flow_of_Dbler_Cycles) * dt
INIT Number_of_Dbler_Cycles = .001

INFLOWS:
Flow_of_Dbler_Cycles = if (Doubler_Deliveries > 0) then
 1
else
```

```
 0
INFLOW TO:
Dbler_Order_Time_Adjust = if (Dbler_Priority=0) then
  Supplier_SOS_Time_Fraction
else
 if (Dbler_Priority=1) then
   Supplier_Priority_1_Time_Fraction
 else
  1

Dbler_Processing_Mean = 5
Dbler_Processing_Std_Dev = 1
Dbler_Processing_Time =
Normal(Dbler_Processing_Mean,Dbler_Processing_Std_Dev,333)*Dbler_Ord
er_Time_Adjust

Supplier Fabrication - Web
Aggregate_Web_Cycle_Time(t) = Aggregate_Web_Cycle_Time(t - dt) +
(Aggregate_Web_Cycle_Time_Flow) * dt
INIT Aggregate_Web_Cycle_Time = .001

INFLOWS:
Aggregate_Web_Cycle_Time_Flow = if (Web_Deliveries > 0) then
  Web_Cycle_Time
else
 0
INFLOW TO:
Number_of_Web_Cycles(t) = Number_of_Web_Cycles(t - dt) +
(Flow_of_Web_Cycles) * dt
INIT Number_of_Web_Cycles = .001

INFLOWS:
Flow_of_Web_Cycles = if (Web_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Web_Cycle_Time(t) = Web_Cycle_Time(t - dt) + (Web_Time_Increment) *
dt
INIT Web_Cycle_Time = 0

INFLOWS:
Web_Time_Increment = if (Web_Orders>0) then
 1
else
  -Web_Cycle_Time
```

```
INFLOW TO:
Web_Orders(t) = Web_Orders(t - dt) + (Web_Incoming_Orders -
Web_Deliveries - Web_Split_Order_Deliveries) * dt
INIT Web_Orders = 0

INFLOWS:
Web_Incoming_Orders = if (Web_Order_Trigger = 1) then
 Web_Order_Backlog
else
 0
INFLOW TO:
OUTFLOWS:
Web_Deliveries = if (Web_Cycle_Time >= Web_Processing_Time) then
 Web_Orders
else
 0
OUTFLOW FROM:
Web_Split_Order_Deliveries = if (Web_Split_Order_Signal = 1) and
(Web_Split_Order_Qty = 0) then
 round(Web_Split_Order_Qtg_Fraction * Web_Orders)
else
 0
OUTFLOW FROM:
Web_Order_Time_Adjust = if (Web_Priority = 0) then
 Supplier_SOS_Time_Fraction
else
 if (Web_Priority = 1) then
  Supplier_Priority_1_Time_Fraction
 else
  1
Web_Processing_Time =
Normal(Web_ROLT_Avg,Web_ROLT_Std_Dev,444)*Web_Order_Time_A
djust
Web_ROLT_Avg = 10
Web_ROLT_Std_Dev = 2

Ordering Sector - Stringers
Stringer_Order_Backlog(t) = Stringer_Order_Backlog(t - dt) +
(Stringer_Order_Releases - Stringer_Order_Completions) * dt
INIT Stringer_Order_Backlog = 0

INFLOWS:
Stringer_Order_Releases = if ((Stringer_Order_Release = 0) and
(Stringer_Switch = 1) and (Probability_of_Lost_Stringer_Orders = 0)) or
(Stringer_Rush_Order = 1) then
```

```
 Stringer_Order_Qty
else
 0
INFLOW TO:
OUTFLOWS:
Stringer_Order_Completions = if Stringer_Deliveries > 0 then
 Stringer_Order_Backlog
else
 0
OUTFLOW FROM:
Stringer_Order_Release(t) = Stringer_Order_Release(t - dt) +
(Stringer_Release_On_and_Off) * dt
INIT Stringer_Order_Release = 0

INFLOWS:
Stringer_Release_On_and_Off = if ((Stringer_Switch = 1) or
(Stringer_Rush_Order = 1)) and (Stringer_Order_Release = 0) then
 1
else
 if (Stringer_Deliveries > 0) then
 -Stringer_Order_Release
else
 0
INFLOW TO:
Stringer_Order_Status(t) = Stringer_Order_Status(t - dt) +
(Stringer_Order_Toggle) * dt
INIT Stringer_Order_Status = 0

INFLOWS:
Stringer_Order_Toggle = if (Stringer_Order_Releases > 0) then
 1
else
-Stringer_Order_Status
INFLOW TO:
Stringer_Flowtime = 71
Stringer_Order_Qty =
Round((Stringer_Rate_of_Use*Stringer_Time_Span)+.5)
Stringer_Order_Trigger = if (Stringer_Order_Status > 0) then
 step(1,Time)
else
 0
Stringer_Rate_of_Use = 1
Stringer_Switch = if (Stringer_Order_Pt = 1) then
 1
else
 0
```

```
Stringer_Time_Span =
Stringer_Time_Span_Parameter*Stringer_Smoothed_Production_Rate
Stringer_Time_Span_Parameter = 44
EOQ = GRAPH(1/Stringer_Smoothed_Production_Rate)
(0.00, 200), (0.526, 157), (1.05, 130), (1.58, 101), (2.11, 81.0), (2.63, 73.0), (3.16, 67.0),
(3.68, 62.0), (4.21, 58.0), (4.74, 54.0), (5.26, 50.0), (5.79, 47.0), (6.32, 45.0), (6.84, 41.0),
(7.37, 39.0), (7.89, 38.0), (8.42, 37.0), (8.95, 36.0), (9.47, 35.0), (10.0, 34.0)
Probability_of_Lost_Stringer_Orders = GRAPH(Time)
Stringer_Smoothed_Production_Rate = GRAPH(Time +
Days_of_Stringer_Inv_Remaining)
(0.00, 0.25), (9.90, 0.25), ...

Ordering Sector - Doubler
Dbler_Order_Backlog(t) = Dbler_Order_Backlog(t - dt) +
(Dbler_Orders_Released - Dbler_Order_Completions) * dt
INIT Dbler_Order_Backlog = 0

INFLOWS:
Dbler_Orders_Released = if ((Doubler_Switch = 1) and
(Doubler_Order_Release = 0) and (Probability_of_Lost_Doubler_Orders = 0))
or (Dbler_Rush_Order = 1) then
 Doubler_Time_Span_Order_Qty
else
 0
INFLOW TO:
OUTFLOWS:
Dbler_Order_Completions = if Doubler_Deliveries > 0 then
 Dbler_Order_Backlog
else
 0
OUTFLOW FROM:
Dbler_Order_Status(t) = Dbler_Order_Status(t - dt) + (Dbler_Order_Toggle) *
dt
INIT Dbler_Order_Status = 0

INFLOWS:
Dbler_Order_Toggle = if Dbler_Orders_Released>0 then
 1
else
-Dbler_Order_Status
INFLOW TO:
Doubler_Order_Release(t) = Doubler_Order_Release(t - dt) +
(Doubler_Release_On_and_Off) * dt
INIT Doubler_Order_Release = 0

INFLOWS:
```

```
Doubler_Release_On_and_Off = if ((Doubler_Switch = 1) or
(Dbler_Rush_Order = 1)) and (Doubler_Order_Release = 0) then
 1
else
 if (Doubler_Deliveries > 0) then
 -Doubler_Order_Release
else
 0
INFLOW TO:
Dbler_Order_Trigger = if Dbler_Order_Status > 0 then
 step(1,Time) + step(-1,Time + 1)
else
 0
Doubler_Flowtime = 83
Doubler_Rate_of_Use = 1
Doubler_Schedule_Advance = 0
Doubler_Switch = if (Doubler_Order_Pt = 1) then
 1
else
 0
Doubler_Time_Span_Order_Qty =
Round((Doubler_Rate_of_Use*Doubler_Smoothed_Production_Rate*Doub
ler_Time_Span_Parameter)+.5)
Doubler_Time_Span_Parameter = 44
Doubler_Smoothed_Production_Rate = GRAPH(Time +
Days_of_Dbler_Inv_Remaining)
(0.00, 0.25), (9.90, 0.25), ...
Probability_of_Lost_Doubler_Orders = GRAPH(Time)
(0.00, 0.00), (2.20, 0.00), ...

Ordering Sector - Assembly
Lot_Number(t) = Lot_Number(t - dt) + (Lot_Increment) * dt
INIT Lot_Number = 1

INFLOWS:
Lot_Increment = if (SA_Orders_Released > 0) then
 1
else
 0
INFLOW TO:
Order_Delay(t) = Order_Delay(t - dt) + (Delay_Flow) * dt
INIT Order_Delay = 0

INFLOWS:
Delay_Flow = if (SA_Orders_Released > 0) then
 1
```

```
else
 -Order_Delay
INFLOW TO:
Order_Release_Status(t) = Order_Release_Status(t - dt) +
(Late_Order_Release) * dt
INIT Order_Release_Status = 0

INFLOWS:
Late_Order_Release = if (SA_Orders_Released > 0) then
 1
else
 if (SA_Kit_Release=1) then
   -Order_Release_Status
 else
  0
INFLOW TO:
SA_Order_Backlog(t) = SA_Order_Backlog(t - dt) + (SA_Orders_Released -
SA_Order_Completions) * dt
INIT SA_Order_Backlog = 0

INFLOWS:
SA_Orders_Released = if (Web_SA_Switch = 1) and (SA_Order_Release=0)
then
  SA_Time_Span_Order_Qty
else
 0
INFLOW TO:
OUTFLOWS:
SA_Order_Completions = if (Subassembly_Deliveries > 0) then
 SA_Order_Backlog
else
 0
OUTFLOW FROM:
SA_Order_Release(t) = SA_Order_Release(t - dt) +
(SA_Release_On_and_Off) * dt
INIT SA_Order_Release = 0

INFLOWS:
SA_Release_On_and_Off = if (Web_SA_Switch = 1) and (SA_Order_Release
= 0) then
 1
else
 if (Subassembly_Deliveries > 0) then
 -SA_Order_Release
else
 0
```

```
INFLOW TO:
Adjustments = if (Lot_Number = 1) then
 -1
else
if (Lot_Number = 2) then
 -2
else
if (Lot_Number = 3) then
 -10
else
if (Lot_Number = 4) then
 2
else
if (Lot_Number = 5) then
 0
else
if (Lot_Number = 6) then
 0
else
if (Lot_Number = 7) then
 -1
else
if (Lot_Number = 8) then
 -1
else
if (Lot_Number = 9) then
 -3
else
if (Lot_Number = 10) then
 -3
else
if (Lot_Number = 11) then
 0
else
if (Lot_Number = 12) then
 0
else
if (Lot_Number = 13) then
 0
else
if (Lot_Number = 14) then
 -3
else
if (Lot_Number = 15) then
 -2
else
```

```
if (Lot_Number = 16) then
 -4
else
if (Lot_Number = 17) then
 0
else
if (Lot_Number = 18) then
 0
else
if (Lot_Number = 19) then
 -5
else
if (Lot_Number = 20) then
 -4
else
if (Lot_Number = 21) then
 0
else
 if (Lot_Number = 22) then
 0
else
 0
Clip_Q_Ck = if (PCA_370_Clip_Stores >= SA_Time_Span_Order_Qty) then
 1
else
 0
Doubler_Q_Ck = if (PCA_370_Doubler_Stores >=
SA_Time_Span_Order_Qty)  then
 1
else
 0
SA_Kit_Check = if Clip_Q_Ck and Web_Q_Ck and Stringer_Q_Ck and
Doubler_Q_Ck  then
 1
else
 0
SA_Kit_Release = if ((SA_Kit_Check = 1) and (Order_Delay =1)) or
((SA_Kit_Check = 1) and (Order_Release_Status >= 1)) or
((SA_Split_Order_Kit_Release = 1) and (Order_Release_Status >= 1)) then
 Step(1,time)
else
 0
SA_Rate_of_Use  =  1
SA_Time_Series_Parameter = if (Lot_Number = 1) then
 20
else
```

```
if (Lot_Number = 2) then
 38
else
if (Lot_Number = 3) then
 80
else
if (Lot_Number = 4) then
 34
else
if (Lot_Number = 5) then
 42
else
if (Lot_Number = 6) then
 45
else
if (Lot_Number = 7) then
 30
else
if (Lot_Number = 8) then
 21
else
if (Lot_Number = 9) then
 39
else
if (Lot_Number = 10) then
 51
else
if (Lot_Number = 11) then
 46
else
if (Lot_Number = 12) then
 45
else
if (Lot_Number = 13) then
 45
else
if (Lot_Number = 14) then
 45
else
if (Lot_Number = 15) then
 48
else
if (Lot_Number = 16) then
 45
else
if (Lot_Number = 17) then
```

```
 25
else
if (Lot_Number = 18) then
 45
else
if (Lot_Number = 19) then
 85
else
if (Lot_Number = 20) then
 85
else
if (Lot_Number = 21) then
 5
else
 if (Lot_Number = 22) then
 45
else
 45
SA_Time_Span_Order_Qty =
Round((SA_Rate_of_Use*SA_Smoothed_Production_Rate*SA_Time_Spa
n_Parameter)+.5)
SA_Time_Span_Parameter = 21
Stringer_Q_Ck = if (PCA_370_Stringer_Stores >=
SA_Time_Span_Order_Qty)  then
 1
else
 0
Subassembly_Flowtime = 10
Web_Q_Ck = if (PCA_370_Web_Stores >= SA_Time_Span_Order_Qty) then
 1
else
 0
Web_SA_Switch = If (Days_of_Web_SA_Inventory_Remaining <=
Subassembly_Flowtime) or (Actual_Days_of_SA_INV_Remaining <
CC352_Expedite_Threshold) then
 1
else
 0
SA_Smoothed_Production_Rate = GRAPH(Time +
Days_of_Web_SA_Inventory_Remaining)
(0.00, 0.25), (9.90, 0.25), ...
```

Ordering Sector - Component Ledger Balances

```
Doublers_on_Ledger_Balance(t) = Doublers_on_Ledger_Balance(t - dt) +
(Additions_to_Doubler_Balance - Deletions_to_Doubler_Balance -
Modifications_to_Doubler_Balance) * dt
```

```
INIT Doublers_on_Ledger_Balance = PCA_370_Doubler_Stores

INFLOWS:
Additions_to_Doubler_Balance = Round(Doubler_Deliveries *
normal(Recording_Accuracy_Mean,Recording_Accuracy_Std_Dev,22))
INFLOW TO:
OUTFLOWS:
Deletions_to_Doubler_Balance = Production_Rate
OUTFLOW FROM:
Modifications_to_Doubler_Balance = if (Dbler_Priority = 0) then
 Doublers_on_Ledger_Balance-PCA_370_Doubler_Stores
else if not (Doubler_Adjustment_Rate = 0) then
 Doubler_Adjustment_Rate
else
 0
OUTFLOW FROM:
Stringers_on_Ledger_Balance(t) = Stringers_on_Ledger_Balance(t - dt) +
(Additions_to_Stringer_Balance - Deletions_to_Stringer_Balance -
Modifications_to_Stringer_Balance) * dt
INIT Stringers_on_Ledger_Balance = PCA_370_Stringer_Stores

INFLOWS:
Additions_to_Stringer_Balance = Round(Stringer_Deliveries *
normal(Recording_Accuracy_Mean,Recording_Accuracy_Std_Dev,11))
INFLOW TO:
OUTFLOWS:
Deletions_to_Stringer_Balance = Production_Rate
OUTFLOW FROM:
Modifications_to_Stringer_Balance = if  (Stringer_Priority = 0) then
 Stringers_on_Ledger_Balance-PCA_370_Stringer_Stores
else if not (Stringer_Adjustment_Rate = 0) then
 Stringer_Adjustment_Rate
else
 0

OUTFLOW FROM:
Days_of_Dbler_Inv_Remaining =
INT((Doublers_on_Ledger_Balance/Smoothed_Production_Rate)+.5)
Days_of_Stringer_Inv_Remaining =
INT((Stringers_on_Ledger_Balance/Smoothed_Production_Rate)+.5)

Ordering Sector - Clips
Clip_Order_Backlog(t) = Clip_Order_Backlog(t - dt) + (Clip_Orders_Released
- Clip_Order_Completions) * dt
INIT Clip_Order_Backlog = 0
```

```
INFLOWS:
Clip_Orders_Released = if ((Clip_Switch = 1) and (Clip_Order_Release = 0)
and (Probability_of_Lost_Clip_Orders = 0)) or (Clip_Rush_Order = 1) then
 Clip_Time_Span_Order_Qty
else
 0
INFLOW TO:
OUTFLOWS:
Clip_Order_Completions = if Clip_Deliveries > 0 then
 Clip_Order_Backlog
else
 0
OUTFLOW FROM:
Clip_Order_Release(t) = Clip_Order_Release(t - dt) +
(Clip_Release_On_and_Off) * dt
INIT Clip_Order_Release = 0

INFLOWS:
Clip_Release_On_and_Off = if ((Clip_Switch = 1) or (Clip_Rush_Order = 1))
and (Clip_Order_Release = 0) then
 1
else
 if (Clip_Deliveries > 0) then
 -Clip_Order_Release
else
 0
INFLOW TO:
Clip_Order_Status(t) = Clip_Order_Status(t - dt) + (Clip_Order_Toggle) * dt
INIT Clip_Order_Status = 0

INFLOWS:
Clip_Order_Toggle = if Clip_Orders_Released>0 then
 1
else
-Clip_Order_Status
INFLOW TO:
Clip_Order_Trigger = if Clip_Order_Status > 0 then
 step(1,Time) + step(-1,Time + 1)
else
 0
Clip_Rate_of_Use = 1
Clip_Schedule_Advance = 42
Clip_Switch = if (Clip_Order_Pt = 1) then
 1
else
 0
```

Clip_Time_Span_Order_Qty = Round((Clip_Rate_of_Use*Clip_Smoothed_Production_Rate*Clip_Time_Span_Parameter)+.5)
Clip_Time_Span_Parameter = 84
ROLT_for_Clip = 180
Clip_Smoothed_Production_Rate = GRAPH(Time + Days_of_Clip_Inv_Remaining)
(0.00, 0.25), (9.90, 0.25), ...
Probability_of_Lost_Clip_Orders = GRAPH(Time)

Ordering System Sector - Rush Orders
Clip_Rush_Order = if (Days_of_Web_SA_Inventory_Remaining < 85) and (Clip_Q_Ck = 0) and (Clip_Order_Backlog = 0) then
 1
else
 0
Dbler_Rush_Order = if (Days_of_Web_SA_Inventory_Remaining < 85) and (Doubler_Q_Ck = 0) and (Dbler_Order_Backlog = 0) then
 1
else
 0
Stringer_Rush_Order = if (Days_of_Web_SA_Inventory_Remaining < 85) and (Stringer_Q_Ck = 0) and (Stringer_Order_Backlog = 0) then
 1
else
 0
Web_Rush_Order = if (Days_of_Web_SA_Inventory_Remaining < 85) and (Web_Q_Ck = 0) and (Web_Order_Backlog = 0) then
 1
else
 0

Ordering System Sector - Order Priority
Total_Clip_Priority_1_Time(t) = Total_Clip_Priority_1_Time(t - dt) + (Clip_Priority_1_Counter) * dt
INIT Total_Clip_Priority_1_Time = 0

INFLOWS:
Clip_Priority_1_Counter = if (Clip_Priority = 1) then
 1
else
 0
INFLOW TO:
Total_Clip_SOS_Time(t) = Total_Clip_SOS_Time(t - dt) + (Clip_SOS_Counter) * dt
INIT Total_Clip_SOS_Time = 0

```
INFLOWS:
Clip_SOS_Counter = if (Clip_Priority = 0) then
 1
else
 0
INFLOW TO:
Total_Dbler_Priority_1_Time(t) = Total_Dbler_Priority_1_Time(t - dt) +
(Dbler_Priority_1_Counter) * dt
INIT Total_Dbler_Priority_1_Time = 0

INFLOWS:
Dbler_Priority_1_Counter = if (Dbler_Priority = 1) then
 1
else
 0
INFLOW TO:
Total_Dbler_SOS_Time(t) = Total_Dbler_SOS_Time(t - dt) +
(Dbler_SOS_Counter) * dt
INIT Total_Dbler_SOS_Time = 0

INFLOWS:
Dbler_SOS_Counter = if (Dbler_Priority = 0) then
 1
else
 0
INFLOW TO:
Total_SA_Priority_1_Time(t) = Total_SA_Priority_1_Time(t - dt) +
(SA_Priority_1_Counter) * dt
INIT Total_SA_Priority_1_Time = 0

INFLOWS:
SA_Priority_1_Counter = if (SA_Priority = 1) then
 1
else
 0
INFLOW TO:
Total_SA_SOS_Time(t) = Total_SA_SOS_Time(t - dt) + (SA_SOS_Counter)
* dt
INIT Total_SA_SOS_Time = 0

INFLOWS:
SA_SOS_Counter = if (SA_Priority = 0) then
 1
else
 0
```

```
INFLOW TO:
Total_Stringer_Priority_1_Time(t) = Total_Stringer_Priority_1_Time(t - dt)
+ (Stringer_Priority_1_Counter) * dt
INIT Total_Stringer_Priority_1_Time = 0

INFLOWS:
Stringer_Priority_1_Counter = if (Stringer_Priority = 1) then
 1
else
 0
INFLOW TO:
Total_Stringer_SOS_Time(t) = Total_Stringer_SOS_Time(t - dt) +
(Stringer_SOS_Counter) * dt
INIT Total_Stringer_SOS_Time = 0

INFLOWS:
Stringer_SOS_Counter = if (Stringer_Priority = 0) then
 1
else
 0
INFLOW TO:
Total_Web_Priority_1_Time(t) = Total_Web_Priority_1_Time(t - dt) +
(Web_Priority_1_Counter) * dt
INIT Total_Web_Priority_1_Time = 0

INFLOWS:
Web_Priority_1_Counter = if (Web_Priority = 1) then
 1
else
 0
INFLOW TO:
Total_Web_SOS_Time(t) = Total_Web_SOS_Time(t - dt) +
(Web_SOS_Counter) * dt
INIT Total_Web_SOS_Time = 0

INFLOWS:
Web_SOS_Counter = if (Web_Priority = 0) then
 1
else
 0
INFLOW TO:
Clip_Priority = if (SA_Order_Release= 1) and (Clip_Q_Ck = 0) and
(SA_Kit_In_Progress = 0) then
 0
else
```

```
 if (Days_of_Clip_Inv_Remaining < (Subassembly_Flowtime + 13)) and
(Clip_Q_Ck = 0) then
 1
 else
 7
Dbler_Priority = if (SA_Order_Release = 1) and (Doubler_Q_Ck = 0) and
(SA_Kit_In_Progress=0)  then
 0
else
 if (Days_of_Dbler_Inv_Remaining < (Subassembly_Flowtime + 13)) and
(Doubler_Q_Ck = 0) then
 1
 else
 7
Lot_Time_Priority_1_Time_Fraction  =  .95
Lot_Time_SOS_Time_Fraction  =  .9
SA_Priority  =  if  (PCA_46_Web_SA_Stores  <=  0)  then
 0
else
if  (Days_of_Web_SA_Inventory_Remaining  <=  13)  then
 1
else
 7
Stringer_Priority = if (SA_Order_Release= 1) and (Stringer_Q_Ck = 0) and
(SA_Kit_In_Progress=0)  then
 0
else
 if (Days_of_Stringer_Inv_Remaining < (Subassembly_Flowtime + 13)) and
(Stringer_Q_Ck = 0) then
 1
 else
 7
Stringer_Priority_1_Time_Fraction  =  .9
Stringer_SOS_Time_Fraction  =  .8
Supplier_Priority_1_Time_Fraction  =  .9
Supplier_SOS_Time_Fraction  =  .8
Web_Priority = if (SA_Order_Release = 1) and (Web_Q_Ck = 0) and
(SA_Kit_In_Progress=0)  then
 0
else
 if (Days_of_Web_Inv_Remaining < (Subassembly_Flowtime + 13)) and
(Web_Q_Ck = 0) then
 1
 else
 7
```

```
Supplier Fabrication - Clip
Aggregate_Clip_Cycle_Time(t) = Aggregate_Clip_Cycle_Time(t - dt) +
(Aggregate_Clip_Cycle_Time_Flow) * dt
INIT Aggregate_Clip_Cycle_Time = .001

INFLOWS:
Aggregate_Clip_Cycle_Time_Flow = if (Clip_Deliveries > 0) then
 Clip_Cycle_Time
else
 0
INFLOW TO:
Clip_Cycle_Time(t) = Clip_Cycle_Time(t - dt) + (Clip_Time_Increment) * dt
INIT Clip_Cycle_Time = 0

INFLOWS:
Clip_Time_Increment = if (Clip_Orders>0) then
 1
else
 -Clip_Cycle_Time

INFLOW TO:
Clip_Orders(t) = Clip_Orders(t - dt) + (Clip_Incoming_Orders -
Clip_Deliveries - Clip_Split_Order_Deliveries) * dt
INIT Clip_Orders = 0

INFLOWS:
Clip_Incoming_Orders = if (Clip_Order_Trigger = 1) then
 Clip_Order_Backlog
else
 0
INFLOW TO:
OUTFLOWS:
Clip_Deliveries = if (Clip_Cycle_Time >= Clip_Processing_Time) then
 Clip_Orders
else
 0
OUTFLOW FROM:
Clip_Split_Order_Deliveries = if (Clip_Split_Order_Signal = 1) and
(Clip_Split_Order_Qty = 0) then
 round(Clip_Split_Order_Qtg_Fraction * Clip_Orders)
else
 0
OUTFLOW FROM:
Number_of_Clip_Cycles(t) = Number_of_Clip_Cycles(t - dt) +
(Flow_of_Clip_Cycles) * dt
INIT Number_of_Clip_Cycles = .001
```

```
INFLOWS:
Flow_of_Clip_Cycles = if (Clip_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Clip_Order_Time_Adjust = if (Clip_Priority = 0) then
  Supplier_SOS_Time_Fraction
else
 if (Clip_Priority=1) then
   Supplier_Priority_1_Time_Fraction
 else
  1

Clip_Processing_Time =
Normal(Clip_ROLT_Avg,Clip_ROLT_Std_Dev,555)*Clip_Order_Time_Adj
ust
Clip_ROLT_Avg = 10
Clip_ROLT_Std_Dev = 2

Ordering Sector - Web
Web_Order_Backlog(t) = Web_Order_Backlog(t - dt) +
(Web_Orders_Released - Web_Order_Completions) * dt
INIT Web_Order_Backlog = 0

INFLOWS:
Web_Orders_Released = if ((Web_Switch = 1) and (Web_Order_Release = 0)
and (Probability_of_Lost_Web_Orders = 0)) or (Web_Rush_Order = 1) then
  Web_Time_Span_Order_Qty
else
 0
INFLOW TO:
OUTFLOWS:
Web_Order_Completions = if Web_Deliveries > 0 then
 Web_Order_Backlog
else
 0
OUTFLOW FROM:
Web_Order_Release(t) = Web_Order_Release(t - dt) +
(Web_Release_On_and_Off) * dt
INIT Web_Order_Release = 0

INFLOWS:
Web_Release_On_and_Off = if ((Web_Switch = 1) or (Web_Rush_Order =
1)) and (Web_Order_Release = 0) then
```

```
 1
else
 if (Web_Deliveries > 0) then
 -Web_Order_Release
else
 0
INFLOW TO:
Web_Order_Status(t) = Web_Order_Status(t - dt) + (Web_Order_Toggle) * dt
INIT Web_Order_Status = 0

INFLOWS:
Web_Order_Toggle = if Web_Orders_Released>0 then
 1
else
-Web_Order_Status
INFLOW TO:
ROLT_of_Web = 180
Web_Order_Trigger = if Web_Order_Status > 0 then
 step(1,Time) + step(-1,Time + 1)
else
 0
Web_Rate_of_Use = 1
Web_Schedule_Advance = 42
Web_Switch = if (Web_Order_Pt = 1) then
 1
else
 0
Web_Time_Span_Order_Qty =
Round((Web_Rate_of_Use*Web_Time_Span_Parameter*Web_Smoothed_
Production_Rate)+.5)
Web_Time_Span_Parameter = 84
Probability_of_Lost_Web_Orders = GRAPH(Time)
Web_Smoothed_Production_Rate = GRAPH(Time +
Days_of_Web_Inv_Remaining)
(0.00, 0.25), (9.90, 0.25), (19.8, 0.25), (29.7, 0.25), (39.6, 0.25), (49.5, 0.25), (59.4, 0.25),
(69.3, 0.25), (79.2, 0.25), ...

Ordering Sector - Component Ledger Balances II
Clips_on_Ledger_Balance(t) = Clips_on_Ledger_Balance(t - dt) +
(Additions_to_Clip_Balance - Deletions_to_Clip_Balance -
Modifications_to_Clip_Balance) * dt
INIT Clips_on_Ledger_Balance = PCA_370_Clip_Stores

INFLOWS:
Additions_to_Clip_Balance = Round(Clip_Deliveries *
normal(Recording_Accuracy_Mean,Recording_Accuracy_Std_Dev,33))
```

```
INFLOW TO:
OUTFLOWS:
Deletions_to_Clip_Balance = Production_Rate
OUTFLOW FROM:
Modifications_to_Clip_Balance = if (Clip_Priority = 0) then
 Clips_on_Ledger_Balance-PCA_370_Clip_Stores
else if not (Clip_Adjustment_Rate = 0) then
 Clip_Adjustment_Rate
else
 0
OUTFLOW FROM:
Webs_on_Ledger_Balance(t) = Webs_on_Ledger_Balance(t - dt) +
(Additions_to_Web_Balance - Deletions_to_Web_Balance -
Modifications_to_Web_Balance) * dt
INIT Webs_on_Ledger_Balance = PCA_370_Web_Stores

INFLOWS:
Additions_to_Web_Balance = Round(Web_Deliveries *
normal(Recording_Accuracy_Mean,Recording_Accuracy_Std_Dev,44))
INFLOW TO:
OUTFLOWS:
Deletions_to_Web_Balance = Production_Rate
OUTFLOW FROM:
Modifications_to_Web_Balance = if (Web_Priority = 0) then
 Webs_on_Ledger_Balance-PCA_370_Web_Stores
else if not (Web_Adjustment_Rate = 0) then
 Web_Adjustment_Rate
else
 0
OUTFLOW FROM:
Days_of_Clip_Inv_Remaining =
INT((Clips_on_Ledger_Balance/Smoothed_Production_Rate)+.5)
Days_of_Web_Inv_Remaining =
INT((Webs_on_Ledger_Balance/Smoothed_Production_Rate)+.5)
```

Ordering Sector - Ledger Balances III

```
Web_SAs_on_Ledger_Balance(t) = Web_SAs_on_Ledger_Balance(t - dt) +
(Additions_to_Web_SA_Balance - Deletions_to_Web_SA_Balance -
Modifications_to_Web_SA_Balance) * dt
INIT Web_SAs_on_Ledger_Balance = PCA_46_Web_SA_Stores

INFLOWS:
Additions_to_Web_SA_Balance = if (Web_SAs_on_Ledger_Balance < 0)
then
```

```
Round(Subassembly_Deliveries *
normal(Recording_Accuracy_Mean,Recording_Accuracy_Std_Dev,55)) +
Web_SAs_on_Ledger_Balance
else
 Round(Subassembly_Deliveries *
normal(Recording_Accuracy_Mean,Recording_Accuracy_Std_Dev,55))

INFLOW TO:
OUTFLOWS:
Deletions_to_Web_SA_Balance = Production_Rate
OUTFLOW FROM:
Modifications_to_Web_SA_Balance = if (SA_Priority = 0) then
  Web_SAs_on_Ledger_Balance-PCA_46_Web_SA_Stores
else if (Web_SA_Adjustment_Rate > 0) then
  Web_SA_Adjustment_Rate
else
 0
OUTFLOW FROM:
Days_of_Web_SA_Inventory_Remaining =
INT((Web_SAs_on_Ledger_Balance/Smoothed_Production_Rate)+.5)
Recording_Accuracy_Mean = 1
Recording_Accuracy_Std_Dev = 0

Lot Time - Split Orders
SA_Kit_In_Progress(t) = SA_Kit_In_Progress(t - dt) +
(SA_Kit_In_Progress_On_Off) * dt
INIT SA_Kit_In_Progress = 0

INFLOWS:
SA_Kit_In_Progress_On_Off = if SA_Kit_Release = 1 then
 1
else
 if Subassembly_Deliveries > 0 then
   -SA_Kit_In_Progress
else
 0
INFLOW TO:
Total_SA_Split_Order_Actions(t) = Total_SA_Split_Order_Actions(t - dt) +
(SA_Split_Order_Counter) * dt
INIT Total_SA_Split_Order_Actions = 0

INFLOWS:
SA_Split_Order_Counter = if (SA_Split_Order_Kit_Release = 1) then
 1
else
 0
```

INFLOW TO:
Days_Until_Next_Delivery = Max(Days_Until_Clip_Delivery,Days_Until_Dbler_Delivery,Days_Until_Stringer_Delivery,Days_Until_Web_Delivery)
Kits_Elements_Short = Max(0,SA_Order_Backlog-Maximum_Available_Kit_Elements)
Kit_Split_Order_Fraction = 1
Kit_Split_Order_Quantity = Maximum_Available_Kit_Elements * Kit_Split_Order_Fraction
Maximum_Available_Kit_Elements = Min(PCA_370_Clip_Stores,PCA_370_Stringer_Stores,PCA_370_Web_Stores,PCA_370_Doubler_Stores)
SA_Split_Order_Kit_Release = if (Split_Order_Release_Trigger = 1) and (SA_Kit_In_Progress = 0) and (Kits_Elements_Short > 0) and (Kit_Split_Order_Quantity > 0) then
1
else
0
Split_Order_Release_Trigger = if ((Days_of_Web_SA_Inventory_Remaining-10) < Days_Until_Next_Delivery) or (Actual_Days_of_SA_INV_Remaining < CC352_Expedite_Threshold) then
1
else
0

Lot Time - Expediting - Days Until Next Delivery
Days_Until_Clip_Delivery = if (Clip_Q_Ck = 0) then
Clip_Processing_Time - Clip_Cycle_Time
else
0
Days_Until_Dbler_Delivery = if (Doubler_Q_Ck = 0) then
Dbler_Processing_Time - Dbler_Cycle_Time
else
0
Days_Until_Stringer_Delivery = if (Stringer_Q_Ck = 0) then
max(0,Stringer_Processing_Time - Stringer_Cycle_Time)
else
0
Days_Until_Web_Delivery = if (Web_Q_Ck = 0) then
Web_Processing_Time - Web_Cycle_Time
else
0

SMC & Suppliers - Expediting

```
Clip_Split_Order_Signal(t) = Clip_Split_Order_Signal(t - dt) +
(Clip_Split_Order_Reset) * dt
INIT Clip_Split_Order_Signal = 0

INFLOWS:
Clip_Split_Order_Reset = if (Web_SA_Switch = 1) and
(PCA_370_Clip_Stores <= Split_Order_Qty_Threshold) and
(Days_Until_Clip_Delivery > Clip_Split_Order_Threshold) and
(Clip_Split_Order_Qty = 0) and (SA_Kit_In_Progress = 0) and
(Clip_Split_Order_Signal = 0) then
 1
else
if (Clip_Split_Order_Qty > 0) then
 -Clip_Split_Order_Signal
else
 0
INFLOW TO:
Doubler_Split_Order_Signal(t) = Doubler_Split_Order_Signal(t - dt) +
(Doubler_Split_Order_Reset) * dt
INIT Doubler_Split_Order_Signal = 0

INFLOWS:
Doubler_Split_Order_Reset = if (Web_SA_Switch = 1) and
(PCA_370_Doubler_Stores <= Split_Order_Qty_Threshold) and
(Days_Until_Dbler_Delivery > Doubler_Split_Order_Threshold) and
(Doubler_Split_Order_Signal = 0) and (Doubler_Split_Order_Qty = 0) and
(SA_Kit_In_Progress = 0) then
 1
else
if (Doubler_Split_Order_Qty > 0) then
 -Doubler_Split_Order_Signal
else
 0
INFLOW TO:
Stringer_Split_Order_Signal(t) = Stringer_Split_Order_Signal(t - dt) +
(Stringer_Split_Order_Reset) * dt
INIT Stringer_Split_Order_Signal = 0

INFLOWS:
Stringer_Split_Order_Reset = if (Web_SA_Switch = 1) and
(PCA_370_Stringer_Stores <= Split_Order_Qty_Threshold) and
(Days_Until_Stringer_Delivery > Stringer_Split_Order_Threshold) and
(Stringer_Split_Order_Signal = 0) and (Stringer_Split_Order_Qty = 0) and
(SA_Kit_In_Progress = 0) then
 1
else
```

```
 if (Stringer_Split_Order_Qty > 0) then
  -Stringer_Split_Order_Signal
else
 0
INFLOW TO:
Web_Split_Order_Signal(t) = Web_Split_Order_Signal(t - dt) +
(Web_Split_Order_Reset) * dt
INIT Web_Split_Order_Signal = 0

INFLOWS:
Web_Split_Order_Reset = if (Web_SA_Switch = 1) and
(PCA_370_Web_Stores <= Split_Order_Qty_Threshold) and
(Days_Until_Web_Delivery > Web_Split_Order_Threshold) and
(Web_Split_Order_Qty = 0) and (SA_Kit_In_Progress = 0) and
(Web_Split_Order_Signal = 0) then
 1
else
if (Web_Split_Order_Qty > 0) then
 -Web_Split_Order_Signal
else
 0
INFLOW TO:
Clip_Split_Order_Qtg_Fraction = .25
Clip_Split_Order_Threshold = 10
Doubler_Split_Order_Qtg_Fraction = .25
Doubler_Split_Order_Threshold = 10
Stringer_Split_Order_Qty_Fraction = .25
Stringer_Split_Order_Threshold = 10
Web_Split_Order_Qtg_Fraction = .25
Web_Split_Order_Threshold = 10

Fabrication - Stringer - Split Order
Stringer_Split_Order_Qty(t) = Stringer_Split_Order_Qty(t - dt) +
(Stringer_Split_Order_Flow - Stringer_Split_Order_Delivery) * dt
INIT Stringer_Split_Order_Qty = 0

INFLOWS:
Stringer_Split_Order_Flow = Stringer_Split_Order_Deliveries
INFLOW TO:
OUTFLOWS:
Stringer_Split_Order_Delivery = if (Stringer_Split_Order_Timer >=
Stringer_Split_Order_Flowtime) then
 Stringer_Split_Order_Qty
else
 0
OUTFLOW FROM:
```

```
Stringer_Split_Order_Timer(t) = Stringer_Split_Order_Timer(t - dt) +
(Stringer_Split_Order_Time_Flow) * dt
INIT Stringer_Split_Order_Timer = 0

INFLOWS:
Stringer_Split_Order_Time_Flow = if (Stringer_Split_Order_Qty > 0) then
 1
else
  -Stringer_Split_Order_Timer
INFLOW TO:
Total_Stringer_Split_Orders(t) = Total_Stringer_Split_Orders(t - dt) +
(Stringer_Split_Order_Counter) * dt
INIT Total_Stringer_Split_Orders = 0

INFLOWS:
Stringer_Split_Order_Counter = if (Stringer_Split_Order_Deliveries > 0)
then
 1
else
 0
INFLOW TO:
Stringer_Split_Order_Flowtime = 5

Fabrication - Suppliers - Split Order Doublers & Webs
Doubler_Split_Order_Qty(t) = Doubler_Split_Order_Qty(t - dt) +
(Doubler_Split_Order_Flow - Doubler_Split_Order_Delivery) * dt
INIT Doubler_Split_Order_Qty = 0

INFLOWS:
Doubler_Split_Order_Flow = Dbler_Split_Order_Deliveries
INFLOW TO:
OUTFLOWS:
Doubler_Split_Order_Delivery = if (Doubler_Split_Order_Timer >=
Doubler_Split_Order_Flowtime) then
 Doubler_Split_Order_Qty
else
 0
OUTFLOW FROM:
Doubler_Split_Order_Timer(t) = Doubler_Split_Order_Timer(t - dt) +
(Doubler_Split_Order_Time_Flow) * dt
INIT Doubler_Split_Order_Timer = 0

INFLOWS:
Doubler_Split_Order_Time_Flow = if (Doubler_Split_Order_Qty > 0) then
 1
else
```

```
 -Doubler_Split_Order_Timer
INFLOW TO:
Total_Doubler_Split_Orders(t) = Total_Doubler_Split_Orders(t - dt) +
(Doubler_Split_Order_Counter) * dt
INIT Total_Doubler_Split_Orders = 0

INFLOWS:
Doubler_Split_Order_Counter = if (Dbler_Split_Order_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Total_Web_Split_Orders(t) = Total_Web_Split_Orders(t - dt) +
(Webs_Split_Order_Counter) * dt
INIT Total_Web_Split_Orders = 0

INFLOWS:
Webs_Split_Order_Counter = if (Web_Split_Order_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Web_Split_Order_Qty(t) = Web_Split_Order_Qty(t - dt) +
(Web_Split_Order_Flow - Web_Split_Order_Delivery) * dt
INIT Web_Split_Order_Qty = 0

INFLOWS:
Web_Split_Order_Flow = Web_Split_Order_Deliveries
INFLOW TO:
OUTFLOWS:
Web_Split_Order_Delivery = if (Web_Split_Order_Timer >=
Web_Split_Order_Flowtime) then
 Web_Split_Order_Qty
else
 0
OUTFLOW FROM:
Web_Split_Order_Timer(t) = Web_Split_Order_Timer(t - dt) +
(Web_Split_Order_Time_Flow) * dt
INIT Web_Split_Order_Timer = 0

INFLOWS:
Web_Split_Order_Time_Flow = if (Web_Split_Order_Qty > 0) then
 1
else
 -Web_Split_Order_Timer
INFLOW TO:
```

```
Doubler_Split_Order_Flowtime = 5
Web_Split_Order_Flowtime = 5

Fabrication - Suppliers - Split Clips
Clip_Split_Order_Qty(t) = Clip_Split_Order_Qty(t - dt) +
(Clip_Split_Order_Flow - Clip_Split_Order_Delivery) * dt
INIT Clip_Split_Order_Qty = 0

INFLOWS:
Clip_Split_Order_Flow = Clip_Split_Order_Deliveries
INFLOW TO:
OUTFLOWS:
Clip_Split_Order_Delivery = if (Clip_Split_Order_Timer >=
Clip_Split_Order_Flowtime) then
 Clip_Split_Order_Qty
else
 0
OUTFLOW FROM:
Clip_Split_Order_Timer(t) = Clip_Split_Order_Timer(t - dt) +
(Clip_Split_Order_Time_Flow) * dt
INIT Clip_Split_Order_Timer = 0

INFLOWS:
Clip_Split_Order_Time_Flow = if (Clip_Split_Order_Qty > 0) then
 1
else
 -Clip_Split_Order_Timer
INFLOW TO:
Total_Clip_Split_Orders(t) = Total_Clip_Split_Orders(t - dt) +
(Clip_Split_Order_Counter) * dt
INIT Total_Clip_Split_Orders = 0

INFLOWS:
Clip_Split_Order_Counter = if (Clip_Split_Order_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Clip_Split_Order_Flowtime = 5

Performance Measurement - Simulated Average Cycle Times
Average_Clip_Cycle_Time =
Round(Aggregate_Clip_Cycle_Time/Number_of_Clip_Cycles)
Average_Doubler_Cycle_Time =
Round(Aggregate_Dbler_Cycle_Time/Number_of_Dbler_Cycles)
```

```
Average_Stringer_Cycle_Time =
Round(Aggregate_Stringer_Cycle_Time/Number_of_Stringer_Cycles)
Average_Subassembly_Cycle_Time =
Round(Aggregate_SA_Cycle_Time/Number_of_SA_Cycles)
Average_Web_Cycle_Time =
Round(Aggregate_Web_Cycle_Time/Number_of_Web_Cycles)
Clip_Priority_1_Cycle_Time_Percentage =
Total_Clip_Priority_1_Time/Aggregate_Clip_Cycle_Time
Clip_SOS_Cycle_Time_Percentage =
Total_Clip_SOS_Time/Aggregate_Clip_Cycle_Time
Doubler_Priority_1_Cycle_Time_Percentage =
Total_Dbler_Priority_1_Time/Aggregate_Dbler_Cycle_Time
Doubler_SOS_Cycle_Time_Percentage =
Total_Dbler_SOS_Time/Aggregate_Dbler_Cycle_Time
Stringer_Priority_1_Cycle_Time_Percentage =
Total_Stringer_Priority_1_Time/Aggregate_Stringer_Cycle_Time
Stringer_SOS_Cycle_Time_Percentage =
Total_Stringer_SOS_Time/Aggregate_Stringer_Cycle_Time
Subassembly_Priority_1_Cycle_Time_Percentage =
Total_SA_Priority_1_Time/Aggregate_SA_Cycle_Time
Subassembly_SOS_Cycle_Time_Percentage =
Total_SA_SOS_Time/Aggregate_SA_Cycle_Time
Web_Priority_1_Cycle_Time_Percentage =
Total_Web_Priority_1_Time/Aggregate_Web_Cycle_Time
Web_SOS_Cycle_Time_Percentage =
Total_Web_SOS_Time/Aggregate_Web_Cycle_Time

Performance Measurement - Actual vs Simulated Aggregate Levels
Doubler_Count(t) = Doubler_Count(t - dt) + (Doubler_Counter) * dt
INIT Doubler_Count = 1

INFLOWS:
Doubler_Counter = if (Doubler_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Historical_Cumulative_Doublers(t) = Historical_Cumulative_Doublers(t -
dt) + (Actual_Doubler_Flow) * dt
INIT Historical_Cumulative_Doublers = PCA_370_Doubler_Stores

INFLOWS:
Actual_Doubler_Flow = Historical_Doubler_Deliveries
INFLOW TO:
Historical_Cumulative_SAs(t) = Historical_Cumulative_SAs(t - dt) +
(Actual_SA_Flow) * dt
```

```
INIT Historical_Cumulative_SAs = PCA_46_Web_SA_Stores

INFLOWS:
Actual_SA_Flow = Historical_Subassembly_Deliveries
INFLOW TO:
Historical_Cumulative_Stringers(t) = Historical_Cumulative_Stringers(t -
dt) + (Actual_Stringer_Flow) * dt
INIT Historical_Cumulative_Stringers = PCA_370_Stringer_Stores

INFLOWS:
Actual_Stringer_Flow = Historical_Stringer_Deliveries
INFLOW TO:
SA_Count(t) = SA_Count(t - dt) + (SA_Counter) * dt
INIT SA_Count = 1

INFLOWS:
SA_Counter = if (Subassembly_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Simulated_Cumulative_Doublers(t) = Simulated_Cumulative_Doublers(t -
dt) + (Sim_Doubler_Flow) * dt
INIT Simulated_Cumulative_Doublers = PCA_370_Doubler_Stores

INFLOWS:
Sim_Doubler_Flow = Doubler_Deliveries
INFLOW TO:
Simulated_Cumulative_Stringers(t) = Simulated_Cumulative_Stringers(t -
dt) + (Sim_Stringer_Flow) * dt
INIT Simulated_Cumulative_Stringers = PCA_370_Stringer_Stores

INFLOWS:
Sim_Stringer_Flow = Stringer_Deliveries
INFLOW TO:
Simulated_Cumulative_Subassemblies(t) =
Simulated_Cumulative_Subassemblies(t - dt) + (Sim_SA_Flow) * dt
INIT Simulated_Cumulative_Subassemblies = PCA_46_Web_SA_Stores

INFLOWS:
Sim_SA_Flow = Subassembly_Deliveries
INFLOW TO:
Stringer_Count(t) = Stringer_Count(t - dt) + (Stringer_Counter) * dt
INIT Stringer_Count = 1

INFLOWS:
```

Stringer_Counter = if (Stringer_Deliveries > 0) then
1
else
0
INFLOW TO:
Avg_Doubler_Order_Qty = round(Simulated_Cumulative_Doublers/Doubler_Count)
Avg_SA_Order_Qty = round(Simulated_Cumulative_Subassemblies/SA_Count)
Avg_Stringer_Order_Qty = round(Simulated_Cumulative_Stringers/Stringer_Count)
Historical_Doubler_Deliveries = step(81,191) + step(-81,192) + step(110,447) + step(-110,448) + step(138,784) + step(-138,785)
Historical_Stringer_Deliveries = step(65,29) + step(-65,30) + step(67,260) + step(-67,261) + step(70,441) + step(-70,442) + step(80,689) + step(-80,690) + step(20,942) + step(-20,943) + step(62,962) + step(-62,963)
Historical_Subassembly_Deliveries = step(4,13) + step(-4,14) + step(7,34) + step(-7,35) + step(10,58) + step(-10,59) + step(10,97) + step(-10,98) + step(14,153) + step(-14,154) + step(15,203) + step(-15,204) + step(9,274) + step(-9,275) + step(6,300) + step(-6,301) + step(11,331) + step(-11,332) + step(13,370) + step(-13,371) + step(15,407) + step(-15,408) + step(15,468) + step(-15,469) + step(15,514) + step(-15,515) + step(12,552) + step(-12,553) + step(14,583) + step(-14,584) + step(14,623) + step(-14,624) + step(10,678) + step(-10,679) + step(18,710) + step(-18,711) + step(29,758) + step(-29,759) + step(30,832) + step(-30,833) + step(2,926) + step(-2,927) + step(18,942) + step(-18,943)

Performance Measurement - Actual vs Simulated Aggregate Levels II
Clip_Count(t) = Clip_Count(t - dt) + (Clip_Counter) * dt
INIT Clip_Count = 1

INFLOWS:
Clip_Counter = if (Clip_Deliveries > 0) then
1
else
0
INFLOW TO:
Historical_Cumulative_Clips(t) = Historical_Cumulative_Clips(t - dt) + (Actual_Clip_Flow) * dt
INIT Historical_Cumulative_Clips = PCA_370_Clip_Stores

INFLOWS:
Actual_Clip_Flow = Historical_Clip_Deliveries
INFLOW TO:
Historical_Cumulative_Webs(t) = Historical_Cumulative_Webs(t - dt) + (Actual_Web_Flow) * dt
INIT Historical_Cumulative_Webs = PCA_370_Web_Stores

```
INFLOWS:
Actual_Web_Flow = Historical_Web_Deliveries
INFLOW TO:
Simulated_Cumulative_Clips(t) = Simulated_Cumulative_Clips(t - dt) +
(Sim_Clip_Flow) * dt
INIT Simulated_Cumulative_Clips = PCA_370_Clip_Stores

INFLOWS:
Sim_Clip_Flow = Clip_Deliveries
INFLOW TO:
Simulated_Cumulative_Webs(t) = Simulated_Cumulative_Webs(t - dt) +
(Sim_Web_Flow) * dt
INIT Simulated_Cumulative_Webs = PCA_370_Web_Stores

INFLOWS:
Sim_Web_Flow = Web_Deliveries
INFLOW TO:
Web_Count(t) = Web_Count(t - dt) + (Web_Counter) * dt
INIT Web_Count = 1

INFLOWS:
Web_Counter = if (Web_Deliveries > 0) then
 1
else
 0
INFLOW TO:
Avg_Clip_Order_Qty = round(Simulated_Cumulative_Clips/Clip_Count)
Avg_Web_Order_Qty = round(Simulated_Cumulative_Webs/Web_Count)
Historical_Clip_Deliveries = step(66,10) + step(-66,11) + step(83,279) + step(-
83,280) + step(96,559) + step(-96,560) + step(88,831) + step(-88,832)
Historical_Web_Deliveries = step(85,128) + step(-85,129) + step(105,409) +
step(-105,410) + step(105,726) + step(-105,727)

Performance Measurement - Stockouts
Clip_SOS_Trigger(t) = Clip_SOS_Trigger(t - dt) + (Clip_SOS_Flow) * dt
INIT Clip_SOS_Trigger = 0

INFLOWS:
Clip_SOS_Flow = if (Clip_SOS_Counter = 1) and (Clip_SOS_Trigger = 0)
then
 1
else
if (Clip_Arrival_Rate > 0) then
 -Clip_SOS_Trigger
else
```

```
 0
INFLOW TO:
Doubler_SOS_Trigger(t) = Doubler_SOS_Trigger(t - dt) +
(Doubler_SOS_Flow) * dt
INIT Doubler_SOS_Trigger = 0

INFLOWS:
Doubler_SOS_Flow = if (Dbler_SOS_Counter = 1) and
(Doubler_SOS_Trigger = 0) then
 1
else
if (Doubler_Arrival_Rate > 0) then
 -Doubler_SOS_Trigger
else
 0
INFLOW TO:
Stringer_SOS_Trigger(t) = Stringer_SOS_Trigger(t - dt) +
(Stringer_SOS_Flow) * dt
INIT Stringer_SOS_Trigger = 0

INFLOWS:
Stringer_SOS_Flow = if (Stringer_SOS_Counter = 1) and
(Stringer_SOS_Trigger = 0) then
 1
else
if (Stringer_Arrival_Rate  > 0) then
 -Stringer_SOS_Trigger
else
 0
INFLOW TO:
Total_Clip_Stockouts(t) = Total_Clip_Stockouts(t - dt) +
(Clip_Stockout_Flow) * dt
INIT Total_Clip_Stockouts = 0

INFLOWS:
Clip_Stockout_Flow = if (Clip_SOS_Flow) then
 1
else
 0
INFLOW TO:
Total_Doubler_Stockouts(t) = Total_Doubler_Stockouts(t - dt) +
(Doubler_Stockout_Flow) * dt
INIT Total_Doubler_Stockouts = 0

INFLOWS:
Doubler_Stockout_Flow = if (Doubler_SOS_Flow = 1) then
```

```
 1
else
 0
INFLOW TO:
Total_Stringer_Stockouts(t) = Total_Stringer_Stockouts(t - dt) +
(Stringer_Stockout_Flow) * dt
INIT Total_Stringer_Stockouts = 0

INFLOWS:
Stringer_Stockout_Flow = if (Stringer_SOS_Flow = 1) then
 1
else
 0
INFLOW TO:
Total_Web_Stockouts(t) = Total_Web_Stockouts(t - dt) +
(Web_Stockout_Flow) * dt
INIT Total_Web_Stockouts = 0

INFLOWS:
Web_Stockout_Flow = if (Web_SOS_Flow = 1) then
 1
else
 0
INFLOW TO:
Web_SOS_Trigger(t) = Web_SOS_Trigger(t - dt) + (Web_SOS_Flow) * dt
INIT Web_SOS_Trigger = 0

INFLOWS:
Web_SOS_Flow = if (Web_SOS_Counter = 1) and (Web_SOS_Trigger = 0)
then
 1
else
if (Web_Arrival_Rate > 0) then
 -Web_SOS_Trigger
else
 0
INFLOW TO:

Performance Measurement - Avg Inventory Level
Total_Clip_Inv(t) = Total_Clip_Inv(t - dt) + (Inv_Flow_4) * dt
INIT Total_Clip_Inv = PCA_370_Clip_Stores

INFLOWS:
Inv_Flow_4 = PCA_370_Clip_Stores
INFLOW TO:
Total_Doubler_Inv(t) = Total_Doubler_Inv(t - dt) + (Inv_Flow_3) * dt
```

```
INIT Total_Doubler_Inv = PCA_370_Doubler_Stores

INFLOWS:
Inv_Flow_3 = PCA_370_Doubler_Stores
INFLOW TO:
Total_SA_Inv(t) = Total_SA_Inv(t - dt) + (Inv_Flow) * dt
INIT Total_SA_Inv = PCA_46_Web_SA_Stores

INFLOWS:
Inv_Flow = PCA_46_Web_SA_Stores
INFLOW TO:
Total_Stringer_Inv(t) = Total_Stringer_Inv(t - dt) + (Inv_Flow_5) * dt
INIT Total_Stringer_Inv = PCA_370_Stringer_Stores

INFLOWS:
Inv_Flow_5 = PCA_370_Stringer_Stores
INFLOW TO:
Total_Web_Inv(t) = Total_Web_Inv(t - dt) + (Inv_Flow_2) * dt
INIT Total_Web_Inv = PCA_370_Web_Stores

INFLOWS:
Inv_Flow_2 = PCA_370_Web_Stores
INFLOW TO:
Avg_Clip_Inv_Level = round(Total_Clip_Inv/Time)
Avg_Doubler_Inv_Level = round(Total_Doubler_Inv/Time)
Avg_SA_Inv_Level = round(Total_SA_Inv/Time)
Avg_Stringer_Inv_Level = round(Total_Stringer_Inv/Time)
Avg_Web_Inv_Level = round(Total_Web_Inv/Time)


Performance Measurement - Expedites
Clip_Expedite_Trigger(t) = Clip_Expedite_Trigger(t - dt) + (Clip_X_Flow) * dt
INIT Clip_Expedite_Trigger = 0

INFLOWS:
Clip_X_Flow = if (Clip_Xpedite_Signal = 1) and (Clip_Expedite_Trigger = 0)
then
 1
else if (Clip_Xpedite_Signal = 0) then
 -Clip_Expedite_Trigger
else
 0
INFLOW TO:
Doubler_Expedite_Trigger(t) = Doubler_Expedite_Trigger(t - dt) +
(Doubler_X_Flow) * dt
INIT Doubler_Expedite_Trigger = 0
```

```
INFLOWS:
Doubler_X_Flow = if (Doubler_Xpedite_Signal =1) and
(Doubler_Expedite_Trigger = 0)  then
 1
else if (Doubler_Xpedite_Signal = 0)  then
 -Doubler_Expedite_Trigger
else
 0
INFLOW TO:
Stringer_Expedite_Trigger(t) = Stringer_Expedite_Trigger(t - dt) +
(Stringer_X_Flow) * dt
INIT Stringer_Expedite_Trigger = 0

INFLOWS:
Stringer_X_Flow = if (Stringer_Xpedite_Signal = 1) and
(Stringer_Expedite_Trigger = 0) then
 1
else if (Stringer_Xpedite_Signal = 0) then
  -Stringer_Expedite_Trigger
else
 0
INFLOW TO:
Total_Clip_Expedites(t) = Total_Clip_Expedites(t - dt) + (Clip_X_Counter) *
dt
INIT Total_Clip_Expedites = 0

INFLOWS:
Clip_X_Counter = if (Clip_X_Flow = 1) then
 1
else
 0
INFLOW TO:
Total_Doubler_Expedites(t) = Total_Doubler_Expedites(t - dt) +
(Doubler_X_Counter) * dt
INIT Total_Doubler_Expedites = 0

INFLOWS:
Doubler_X_Counter = if (Doubler_X_Flow = 1) then
 1
else
 0
INFLOW TO:
Total_Stringer_Expedites(t) = Total_Stringer_Expedites(t - dt) +
(Stringer_X_Counter) * dt
INIT Total_Stringer_Expedites = 0
```

```
INFLOWS:
Stringer_X_Counter = if (Stringer_X_Flow = 1) then
 1
else
 0
INFLOW TO:
Total_Web_Expedites(t) = Total_Web_Expedites(t - dt) + (Web_X_Counter) *
dt
INIT Total_Web_Expedites = 0

INFLOWS:
Web_X_Counter = if (Web_X_Flow = 1) then
 1
else
 0
INFLOW TO:
Web_Expedite_Trigger(t) = Web_Expedite_Trigger(t - dt) + (Web_X_Flow) *
dt
INIT Web_Expedite_Trigger = 0

INFLOWS:
Web_X_Flow = if (Web__Xpedite_Signal = 1) and (Web_Expedite_Trigger =
0) then
 1
else if (Web__Xpedite_Signal = 0)  then
 -Web_Expedite_Trigger
else
 0
INFLOW TO:
Clip_Xpedite_Signal = if ((Days_of_Web_SA_Inventory_Remaining <=
Detail_Xpedite_Threshold) or (Actual_Days_of_SA_INV_Remaining <
CC352_Expedite_Threshold)) and (Clip_Q_Ck = 0) and
(SA_Kit_In_Progress=0)  then
 1
else
 0
Detail_Xpedite_Threshold = 10
Doubler_Xpedite_Signal = if ((Days_of_Web_SA_Inventory_Remaining <=
Detail_Xpedite_Threshold) or (Actual_Days_of_SA_INV_Remaining <
CC352_Expedite_Threshold)) and (Doubler_Q_Ck = 0) and
(SA_Kit_In_Progress=0)  then
 1
else
 0
```

```
Stringer_Xpedite_Signal = if ((Days_of_Web_SA_Inventory_Remaining <=
Detail_Xpedite_Threshold) or (Actual_Days_of_SA_INV_Remaining <
CC352_Expedite_Threshold)) and (Stringer_Q_Ck = 0) and
(SA_Kit_In_Progress=0) then
 1
else
 0
Web__Xpedite_Signal = if ((Days_of_Web_SA_Inventory_Remaining <=
Detail_Xpedite_Threshold) or (Actual_Days_of_SA_INV_Remaining <
CC352_Expedite_Threshold)) and (Web_Q_Ck = 0) and (SA_Kit_In_Progress
= 0) then
 1
else
 0

Performance Measurement - Expedites
SA_Expedite_Trigger(t) = SA_Expedite_Trigger(t - dt) + (SA_X_Flow) * dt
INIT SA_Expedite_Trigger = 0

INFLOWS:
SA_X_Flow = if (SA_Xpedite_Signal = 1) and (SA_Expedite_Trigger = 0)
then
 1
else if (SA_Xpedite_Signal = 0) then
 -SA_Expedite_Trigger
else
 0
INFLOW TO:
Total_SA_Expedites(t) = Total_SA_Expedites(t - dt) + (SA_X_Counter) * dt
INIT Total_SA_Expedites = 0

INFLOWS:
SA_X_Counter = if (SA_X_Flow = 1) then
 1
else
 0
INFLOW TO:
SA_Xpedite_Signal = if (Actual_Days_of_SA_INV_Remaining <
CC352_Expedite_Threshold) then
 1
else
 0

Lot Time - Inventory Loss
CLip_Inv_Stock(t) = CLip_Inv_Stock(t - dt) + (Clip_Inv_Flow) * dt
INIT CLip_Inv_Stock = 0
```

```
INFLOWS:
Clip_Inv_Flow = if (Clip_Arrival_Rate > 0) then
 Clip_Arrival_Rate
else
 -CLip_Inv_Stock

INFLOW TO:
Doubler_Inv_Stock(t) = Doubler_Inv_Stock(t - dt) + (Dcubler_Inv_Flow) *
dt
INIT Doubler_Inv_Stock = 0

INFLOWS:
Doubler_Inv_Flow = if (Doubler_Arrival_Rate > 0) then
 Doubler_Arrival_Rate
else
 -Doubler_Inv_Stock

INFLOW TO:
SA_Inv_Stock(t) = SA_Inv_Stock(t - dt) + (SA_Inv_Flow) * dt
INIT SA_Inv_Stock = 0

INFLOWS:
SA_Inv_Flow = if (Web_SA_Arrival_Rate > 0) then
 Web_SA_Arrival_Rate
else
 -SA_Inv_Stock

INFLOW TO:
Stringer_Inv_Stock(t) = Stringer_Inv_Stock(t - dt) + (Stringer_Inv_Flow) *
dt
INIT Stringer_Inv_Stock = 0

INFLOWS:
Stringer_Inv_Flow = if (Stringer_Arrival_Rate > 0) then
 Stringer_Arrival_Rate
else
 -Stringer_Inv_Stock

INFLOW TO:
Web_Inv_Stock(t) = Web_Inv_Stock(t - dt) + (Web_Inv_Flow) * dt
INIT Web_Inv_Stock = 0

INFLOWS:
Web_Inv_Flow = if (Web_Arrival_Rate > 0) then
 Web_Arrival_Rate
```

```
else
 -Web_Inv_Stock

INFLOW TO:
Clip_Inv_Loss = round(CLip_Inv_Stock*Clip_Loss_Prob)
Clip_Loss_Prob =
normal(Inventory_Loss_Mean,Inventory_Loss_Std_Dev,1111)
Doubler_Inv_Loss = round(Doubler_Inv_Stock*Doubler_Loss_Prob)
Doubler_Loss_Prob =
normal(Inventory_Loss_Mean,Inventory_Loss_Std_Dev,3333)
Inventory_Loss_Mean = 0
Inventory_Loss_Std_Dev = 0
SA_Inv_Loss = round(SA_Inv_Stock*SA_Loss_Prob)

SA_Loss_Prob =
normal(Inventory_Loss_Mean,Inventory_Loss_Std_Dev,5555)
Stringer_Inv_Loss = round(Stringer_Inv_Stock*Stringer_Loss_Prob)
Stringer_Loss_Prob =
normal(Inventory_Loss_Mean,Inventory_Loss_Std_Dev,4444)
Web_Inv_Loss = round(Web_Inv_Stock*Web_Loss_Prob)
Web_Loss_Prob =
normal(Inventory_Loss_Mean,Inventory_Loss_Std_Dev,2222)

Lot Time - Quality Loss
CLip_Inv_Stock_2(t) = CLip_Inv_Stock_2(t - dt) + (Clip_Inv_Flow_2) * dt
INIT CLip_Inv_Stock_2 = 0

INFLOWS:
Clip_Inv_Flow_2 = if (Clip_Arrival_Rate > 0) then
 Clip_Arrival_Rate
else if (Clip_Kit_xfer > 0) then
 -CLip_Inv_Stock_2
else
 0

INFLOW TO:
Doubler_Inv_Stock_2(t) = Doubler_Inv_Stock_2(t - dt) +
(Doubler_Inv_Flow_2) * dt
INIT Doubler_Inv_Stock_2 = 0

INFLOWS:
Doubler_Inv_Flow_2 = if (Doubler_Arrival_Rate > 0) then
 Doubler_Arrival_Rate
else if (Doubler_Kit_xfer > 0) then
 -Doubler_Inv_Stock_2
else
```

```
 0

INFLOW TO:
SA_Inv_Stock_2(t) = SA_Inv_Stock_2(t - dt) + (SA_Inv_Flow_2) * dt
INIT SA_Inv_Stock_2 = 0

INFLOWS:
SA_Inv_Flow_2 = if (Web_SA_Arrival_Rate > 0) then
 Web_SA_Arrival_Rate
else
 -SA_Inv_Stock_2

INFLOW TO:
StringerInv_Stock_2(t) = StringerInv_Stock_2(t - dt) +
(Stringer_Inv__Flow_2) * dt
INIT StringerInv_Stock_2 = 0

INFLOWS:
Stringer_Inv__Flow_2 = if (Stringer_Arrival_Rate > 0) then
 Stringer_Arrival_Rate
else if (Stringer_Kit_xfer > 0) then
 -StringerInv_Stock_2
else
 0

INFLOW TO:
Web_Inv_Stock_2(t) = Web_Inv_Stock_2(t - dt) + (Web_Inv_Flow_2) * dt
INIT Web_Inv_Stock_2 = 0

INFLOWS:
Web_Inv_Flow_2 = if (Web_Arrival_Rate > 0) then
 Web_Arrival_Rate
else if (Web_Kit_xfer > 0) then
 -Web_Inv_Stock_2
else
 0

INFLOW TO:
Clip_Loss_Prob_2 =
normal(Inventory_Q_Loss_Mean,Inventory_Q_Loss_Std_Dev,1111)
Clip_Q_Loss = round(CLip_Inv_Stock_2*Clip_Loss_Prob_2)
Doubler_Loss_Prob_2 =
normal(Inventory_Q_Loss_Mean,Inventory_Q_Loss_Std_Dev,3333)
Doubler_Q_Loss = round(Doubler_Inv_Stock_2*Doubler_Loss_Prob_2)
Inventory_Q_Loss_Mean = 0
Inventory_Q_Loss_Std_Dev = .02
```

```
SA_Loss_Prob_2 =
normal(Inventory_Q_Loss_Mean,Inventory_Q_Loss_Std_Dev,5555)
SA_Q_Loss_2 = round(SA_Inv_Stock_2*SA_Loss_Prob_2)

Stringer_Loss_Prob_2 =
normal(Inventory_Q_Loss_Mean,Inventory_Q_Loss_Std_Dev,4444)
Stringer_Q_Loss = round(StringerInv_Stock_2*Stringer_Loss_Prob_2)
Web_Loss_Prob_2 =
normal(Inventory_Q_Loss_Mean,Inventory_Q_Loss_Std_Dev,2222)
Web_Q_Loss = round(Web_Inv_Stock_2*Web_Loss_Prob_2)

Ordering Sector - Order Pt
Clip_Order_Pt = if (PCA_370_Clip_Stores <=
(Smoothed_Production_Rate*External_Order_Pt)) then
 1
else
 0
Doubler_Order_Pt = if (PCA_370_Doubler_Stores <=
(Smoothed_Production_Rate*Internal_Order_Pt)) then
 1
else
 0
External_Order_Pt = 42
Internal_Order_Pt = 30
Stringer_Order_Pt = if (PCA_370_Stringer_Stores <=
(Smoothed_Production_Rate*Internal_Order_Pt)) then
 1
else
 0
Web_Order_Pt = if (PCA_370_Web_Stores <=
(Smoothed_Production_Rate*External_Order_Pt)) then
 1
else
 0

Not in a sector
Split_Order_Qty_Threshold = 10
```

Appendix A4 - Model Validation

In this appendix, the process of model validation is discussed. The tests used are described and the results reviewed. Three different types of model validation tests were performed: 1) tests of model structure, 2) tests of model behavior, and 3) tests of policy implications.

Tests of Model Structure

Tests of model structure determine the validity of the structure independent of the behavior it produces. Model structure tests are broken down into five categories: 1) structure verification, 2) parameter verification, 3) extreme conditions, 4) boundary adequacy, and 5) dimensional consistency.

Structure Verification

Does the model structure have similar elements seen in the real system ? Is there anything in the model structure that appears foreign to the actual system ? This test was performed by comparison of the model structure to notes collected on the real system during an internship at Boeing. The preliminary structure of the model was also reviewed by experts on the Boeing ordering system prior to the end of the internship, and they found the structure to be consistent with elements of the real system. Unfortunately, time and distance, prevent a thorough review of the final model structure with Boeing employees.

Parameter Verification

Parameters, the constants in the system, should also accurately represent the real values seen in the system. Are the parameters used in the model reasonable based on current knowledge of the system? For example, if the time required for the real system to detect and correct inventory discrepancies is 2 weeks and the model assumes it to occur instantaneously, the model will obviously be in error. The parameter values used, and the sources for those values, can be seen in Table B-3.4 of Appendix B-3. Most of the parameter values have a clear tie to formal policies or other Boeing documentation; some, however, are estimated.

Extreme Conditions

How does the model respond to state variables that vary over extreme regions ? For example, if the stores level in an area reach zero in the model and the model continues to generate product, the model has a structural error because the actual system does not behave in this way. Extreme condition tests were conducted with this model by varying the state variables from very high to very low values. For example, when all inventory is eliminated within storage areas in the model, orders are not generated until the inventory is replenished. Conversely, when the storage areas have an abundance of inventory, orders are also not generated until the existing inventory is depleted.

Boundary Adequacy

The goal of this test is to determine if the model includes all relevant structure ? Is the model aggregated to such a level that important sub-structure is not included ? Is the model boundary set at the right limit including all variables necessary to endogenously generate the reference behavior ? This model depends on an exogenous input for the aircraft production rate. There are many factors which affect the aircraft production rate like macroeconomic conditions, airline financial health, etc., and therefore an attempt to endogenously reproduce the aircraft production rate was not warranted. One area that deserves further study is the ordering policy that is used. The model assumes that the ordering policy is constant (set by Boeing corporate policy guidance), but the ability to vary the policy does reside within the individual production centers. One area of possible model improvement is to attempt to endogenously capture the decision rule used to vary ordering policy from the corporate standard.

Dimensional Consistency

This is a simple test that determines if there is unit consistency throughout the model. A dimensional analysis of the rate equations in the model is performed. For this model, the reader can trace the dimensional consistency by reviewing the equation listing within the software where all of the rate equations are documented.

Tests of Model Behavior

Tests of model behavior determine the validity of the structure by evaluating the behavior it produces in comparison to the real system. Model behavior tests are broken down into two categories: 1) behavior reproduction, and
2) behavior sensitivity.

Behavior Reproduction

How accurately does the model generated behavior match what is experienced in the real system ? Chapter 4 of this thesis reviews the performance of the model in comparison with historical data from the real system. The results of several statistical tests indicate that the model accurately reproduces the reference behavior seen in the Boeing ordering system.

Behavior Sensitivity

How does the model perform when plausible changes in the parameter values are made? With changes in parameter values, there will be shifts in the model response, but the basic behavior of the underlying system should not be lost. The sensitivity tests performed on this model can be seen in Table A-4.1 of this appendix. The parameter tested, range of values used, and the number of simulation runs performed are shown.

Tests of Policy Implications

Tests of policy implications attempt to verify the response of the real system to the corresponding response predicted by the model for various policy changes. Policy implications tests are broken down into two categories: 1) changed-behavior prediction, and 2) policy sensitivity.

Changed-Behavior Prediction

The changed-behavior prediction tests attempt to determine if the model correctly predicts the response of the system when policy changes are attempted. If a policy change is tried in the model and the resulting model response is clearly implausible, the model accuracy is called into question. Different policy actions were simulated with this model, and the results were discussed in Chapter 5. All policy actions attempted produced a reasonable response. For example, lowering inventory levels, without improving the underlying process, leads to increases in order priority levels and component shortages. The performance score sheets document the model response to the various policy changes attempted.

Policy Sensitivity

Is it possible for the model to recommend different policies as parameter values are varied over a reasonable range, or does the model policy recommendation remain consistent over a plausible parameter range? For this model, the response remained consistent as parameter values were varied during the policy analysis in Chapter 5.

Table A-4.1 - Sensitivity Runs

Parameter/Variable	**Range of Values**	**Sensitivity Runs**
CC 352 Production Rate	.05 - 1.95 aircraft/day	10
Assembly Order Qty	1 - 100 day time span	10
Assembly Flowtime	1 - 60 days	5
Assembly Rate of Use	1 - 2	5
Assembly Processing	5 - 50 days	20
Stringer Processing	1 - 100 days	24
Stringer Flowtime	1 - 100 days	5
Stringer Order Qty	1 - 200 units	5
Stringer Rate of Use	1 - 2	5
Doubler Processing	1 - 100 days	20
Doubler Flowtime	1 - 100 days	5
Doubler Order Qty	1 - 200 units	5
Doubler Rate of Use	1 -2	5
Web Processing	100 - 220 days	10
Web ROLT	100 - 220 days	5
Web Order Qty	1 - 504 day time span	5
Clip Processing	100 - 220 days	5
Clip ROLT	100 - 220 days	5
Clip Order Qty	1 - 504 day time span	5
Supplier Schedule Advance	1 - 84 days	5
Rejection Rates	0 - 100 %	10
Order Priority Timing	1 - 30 days from SOS	10

Parameter/Variable	Range of Values	Sensitivity Runs
Split Order Quantities	.05 - .50	5
SOS Time Fraction	.1 - 1	10
Priority 1 Time Fraction	.1 - 1	10
Split Order Threshold (days until next delivery)	1 - 50	10
Split Order Threshold (Qty in Bins)	0 - 10	10
Split Order Flowtimes	1 - 50 days	10
Inventory Accuracy	50-100%	5

Appendix B1 - Discussion of Economic Order Quantity (EOQ) Logic

Boeing orders many internally produced detail components in quantities determined by EOQ logic. Economic order quantities represent a tradeoff between the cost to produce the component versus the cost to hold the component in storage as shown in Table B-1.1 below. The procurement cost consists of an administrative cost, which represents the aggregate cost (i.e. - paper, personnel time, distribution, etc.) to process an order, and the setup cost which is determined by how long it takes to setup the equipment needed to process the component and the labor rate of the person that performs the work. The per day holding cost is based on a per day rate that is multiplied by the unit cost of the component. The holding cost factor is based on several

---

Table B-1.1 - Procurement & Holding Cost Determination

Procurement Cost = Administrative Cost + SetUp Cost

- Administrative Cost = $83.72 {constant value}
- SetUp Cost = SetUp time * Standard Labor Rate
    - SetUp Time = from component Manufacturing Plan
    - Standard Labor Rate = $125.42 per hour

Per Day Holding Cost = Unit Cost * (Holding Cost Factor/Workdays per Year)

Unit Cost = Run Cost + Material Cost

- Run Cost = Run Time * Standard Labor Rate
    - Run Time = from component Manufacturing Plan
    - Standard Labor Rate = $125.42 per hour
- Material Cost = material dependent
- Holding Cost Factor = based on the following considerations
- { Interest, Obsolescence, Storage & Insurance, Records & Handling}
- Current Holding Cost Factor = 19 %
- Workdays per Year = 252

---

considerations, and it is meant to capture how much it costs the company to have inventory sitting in its stores. The unit cost of a component is composed of the run cost and material cost to produce a unit. The run cost is determined by the run time and the standard labor rate. The material cost is dependent on the type of material used, and this information is provided by Materiel Division. An example of an EOQ calculation can be seen in Table B-1.2 below.

Table B-1.2 - Example of EOQ Calculations

Values for Calculation

Set Up Time = 3.17 hours

Unit Cost = $75.51

Production Rate = 1 component used every 3 days

Procurement Cost = Administrative Cost + Setup Cost
= $83.72 + (3.17 hrs * $125.42) = $480.88

Per Day Holding Cost = Unit Cost * (Factor/Workdays per Year)
= $75.51 * (.19/252) = $.0569 per M-day

Units	**Carrying Days**	**Individual Charge ($s)**	**Cumulative Charge ($s)**
1	0	0	0
2	3	.17	.17
3	6	.34	.51
....	....	....	....
75	222	12.63	473.69

There is no holding charge attributed to the first unit of the lot because it is assumed that it will be consumed immediately. Based on the calculation above, the EOQ for this component is 75. If 76 units were produced, the cumulative holding cost would exceed the procurement cost ($480.88) of the component.

Possible Problems with the Use of EOQ for Ordering Action

Quality of Input Data

Several problems exist with EOQ calculations in general, and with this formulation specifically. First of all, many of the values used in these calculations are of questionable accuracy. The administrative cost is a fully burdened cost for each order. Obviously the marginal cost of an additional order is not $83.72. If 20 orders are cut versus 10, there is some additional cost but not $830.72 . The setup and run time values used are also suspect. These times are estimates of what the time should be when the manufacturing plan is developed. The actual times may be radically different and frequently are. The inventory holding cost factor is a value that attempts to capture all the costs associated with holding inventory. One can imagine two extremes in determining this value (neither good). Either hours and hours are spent in an attempt to accurately define this value, or the value is based more on a "gut feel" from current interest rates and other factors. In any event, the factor used is an educated guess. Based on the quality of the input to the algorithm, how reliable can the output really be ?

EOQ Assumptions

A second problem with EOQ calculations has to do with the assumptions that are imbedded within the algorithm. For example, the EOQ logic assumes that

the firs
associa
many
Additi
fact, s
Curren
improv
times
manufa

Resour

Thirdly
are req
differe
the cal
manpo
method

Unseen

Finally,
The a
consum
in a ti
improv
effect.
on the
arrivin

they were constantly in a reactive posture because of their inability to forecast future production requirements.

Appendix B2 - Information on Assembly used in Analysis

The assembly used for analysis is part of the Boeing 757 #2 Passenger door frame and is composed of the part numbers shown in Table B-2.1 below.

Table B-2.1 - Part Numbers used in Assembly

Part Number	Description	Manufacturer
143N8343-904	Assembly	Renton Lot Time
143N3210-4	Stringer	Sheet Metal Center
143N8343-11	Doubler	Sheet Metal Center
143N8343-10	Web	Kaman
143N8353-8	Clip	Anadite

Part Records list key information about the components produced for Boeing aircraft. Tables B-2.2 - B-2.6 below contain a summary of some of the component characteristics taken off of their part records. The increment code groups the component into a cost category. The first letter indicates the type of component: D - detail, A - assembly, U - unit. The second letter represents the cost category, as shown in Table B-2.2 for detail components.

Table B-2.2 - Increment Code Cost Range for Detail Components

Increment	Cost Range
B	.01-10
C	10.01-30
D	30.01-120
E	120.01-250
F	250.01-450
G	450.01-800
H	800.01-over

For assemblies, the increment code is used to determine the time span parameter to apply for lot ordering. Since the assembly used for analysis is a Increment cost code D, a two month time span is used for ordering as indicated in Table B-2.3.

Table B-2.3 - Lot Ordering

Increment	Lot Time Span
B	3 months supply
C-D	2
E or higher	1

Table B-2.4 - Assembly Part Information

Part Information	**Assembly**
Increment Code	AD
Material Code	NA
Unit Cost	106.71
Using Shop	CC 352
End Use Shop	CC 352
Ordering Policy	2 month Time Span
Allocated Flowtime	30 days

The material code is a ten digit code that uniquely identifies the type of material used in the manufacture of the component. The unit cost is the sum of the material cost and the run cost associated with manufacturing the

Table B-2.5 - Stringer & Doubler Part Information

Part Information	**Stringer**	**Doubler**
Increment Code	DD	DB
Material Code	13 6407 1700	11 2402 5601
Unit Cost	75.51	3.08
Using Shop	172	172
End Use Shop	CC 352	CC352
Ordering Policy	EOQ	Order Base
Allocated Flowtime	71 days	83 days

component. The using shop represents the next shop to receive the component. The end use shop, as its name implies, is the last shop to receive the component. For details, they will all be part of a larger assembly by the time they arrive at the end use shop. In the case of the assembly used in the model, all of the details will be assembled at shop 172, and as part of the subassembly, they will then be delivered to Control Code 352 for installation. The ordering policy determines the quantity of component produced. The allocated flowtime is based on the Puget Sound Flowtime and varies depending on the component in question and where it is manufactured.

Table B-2.6 - Web & Clip Part Information

Part Information	**Web**	**Clip**
Increment Code	DC	DC
Material Code	11 2404 0301	11 2404 0301
Unit Cost	13.35	14.77
Using Shop	172	172
End Use Shop	CC 352	CC 352
Ordering Policy	1 year Time Span	1 Year Time Span
Allocated Flowtime	180 days	180 days

Appendix B3 - Samples of Boeing Data used in Analysis

Included in this appendix are summaries of the order activity for the components used in the model. They can be seen in Table B-3.1 and B-3.2. In Table B-3.3, a glossary of all parameter values is provided. The sources used to determine the parameter values are presented in Table B-3.4 .

Table B-3.1 - Assembly Historical Data

Assembly					
Part Number	Order #	Actual Start	Actual Complete	PCA 370 Rec	Qty Delivered
143N8343-904	1	8682	8692	8684	5
143N8343-904	3	8773	8783	8765	8
143N8343-904	2	8731	8741	8773	0
143N8343-904	4	8821	8831	8795	8
143N8343-904	5	8827	8837	8828	8
143N8343-904	6	8869	8879	8886	10
143N8343-904	7	8946	8956	8960	5
143N8343-904	8	8981	8991	8995	5
143N8343-904	9	9050	9060	9067	6
143N8343-904	10	9107	9117	9125	9
143N8343-904	11	9147	9157	9165	9
143N8343-904	12	9192	9202	9209	8
143N8343-904	13	9237	9247	9251	6
143N8343-904	14	9287	9297	9301	6
143N8343-904	15	9342	9352	9353	11
143N8343-904	16	9406	9416	9413	7
143N8343-904	18	9466	9476	9441	0
143N8343-904	20	9502	9512	9462	0
143N8343-904	19	9482	9492	9462	0
143N8343-904	17	9477	9487	9462	0
143N8343-904	21	9477	9487	9501	11
143N8343-904	22	9527	9537	9549	4
143N8343-904	21	9526	9545	9570	7
143N8343-904	23	9571	9581	9594	10
143N8343-904	24	9615	9625	9633	10
143N8343-904	25	9663	9673	9689	14
143N8343-904	26	9705	9715	9739	15
143N8343-904	27	9789	9799	9810	9
143N8343-904	28	9819	9829	9836	6
143N8343-904	29	9840	9850	9867	11
143N8343-904	30	9885	9895	9906	13
143N8343-904	31	9930	9940	9943	15
143N8343-904	32	9976	9986	10004	15
143N8343-904	33	10021	10031	10050	15
143N8343-904	34	10066	10076	10088	12
143N8343-904	35	10111	10121	10119	14
143N8343-904	36	10155	10165	10159	14
143N8343-904	37	10200	10210	10214	10
143N8343-904	38	10225	10235	10246	18

Assembly					
Part Number	Order #	Actual Start	Actual Complete	PCA 370 Rec	Qty Delivered
143N8343-904	39	10270	10280	10294	29
143N8343-904	40	10355	10365	10368	30
143N8343-904	41	10440	10450	10462	2
143N8343-904	42	10445	10455	10478	18

Table B-3.2 - Component Historical Data

STRINGER					
Part Number	Order #	Actual Start	Actual Complete	PCA 370 Rec	Qty Delivered
143N3210-4	16	9464	9507	9536	4
143N3210-4	17	9464	9507	9530	0
143N3210-4	18	9456	9499	9565	65
143N3210-4	19	9755	9798	9796	67
143N3210-4	20	9951	9999	9977	70
143N3210-4	21	10176	10205	10225	80
143N3210-4	22	10397	10425	10478	20
143N3210-4	22A	10395	10425	10498	62
DOUBLER					
Part Number	Order #	Actual Start	Actual Complete	PCA 370 Rec	Qty Delivered
143N8343-11	2	9245	9267	9276	208
143N8343-11	3	9737	9759	9727	162
143N8343-11	4	10029	10062	9983	220
143N8343-11	5	10301	10334	10320	277
CLIP					
Part Number	Order #	Actual Start	Actual Complete	PCA 370 Rec	Qty Delivered
143N8353-8	2	NA	NA	9546	66
143N8353-8	3	NA	NA	9815	83
143N8353-8	4	NA	NA	10095	96
143N8353-8	5	NA	NA	10367	88
WEB					
Part Number	Order #	Actual Start	Actual Complete	PCA 370 Rec	Qty Delivered
143N8343-10	2	NA	NA	9664	85
143N8343-10	3	NA	NA	9945	105
143N8343-10	4	NA	NA	10262	105

Table B-3.3 - Glossary of Parameters

Aircraft Production Rate (aircraft produced every x days)
the rate at which aircraft roll out the doors of final assembly; for the assembly used in this analysis the roll out rate and production rate in control code 352 were identical

Rejection & Rework Rate (percent of order scrapped)
the amount of product loss due to poor quality or engineering change; for the purposes of this model, both rejection and rework are treated as product loss from stores

Stringer Economic Order Quantity Values (units per order)
order quantity determined through use of a Boeing EOQ algorithm

Average Stringer Processing Time (days)
the mean and standard deviation of the processing time for stringer production within the Sheet Metal Center

Puget Sound Flowtime - Stringer (days)
the planning flowtime used when ordering stringers

Doubler Time Span (days)
this parameter determines the order quantity; the number of units needed to satisfy production requirements during this time span will be built

Average Doubler Processing Time (days)
the mean and standard deviation of the processing time for doubler production within Fabrication Division

Puget Sound Flowtime - Doubler (days)
the planning flowtime used when ordering doublers

Component Rate of Use Values (unitless)
a protective ordering factor used when ordering some components; it can vary from 1-1.5; for example, if the rate of use parameter is set at 1.5, then 1.5 times the production requirements will be ordered; this parameter is used to compensate for losses within the system

Priority 1 Time Fraction (percent of normal flow)
when an order has a higher priority, it is worked on first; this parameter estimates the percentage of normal flowtime required to process priority 1 orders

SOS Time Fraction (percent of normal flow)

when an order has a higher priority, it is worked on first; this parameter estimates the percentage of normal flowtime required to process SOS orders

Split Order Quantity Fraction (percent of original order)

when an order is split, this parameter estimates how much of the original order is expedited

Split Order Flowtime (days)

after an order has been split and is being expedited, this parameter estimates the flowtime required to receive the split order

Average Subassembly Processing Time (days)

the mean and standard deviation of the processing time for component assembly within the Renton Lot Time area

Subassembly Time Span (days)

this parameter determines the order quantity; the number of units needed to satisfy production requirements during this time span will be built

Puget Sound Flowtime - Lot Time Subassembly (days)

the planning flowtime used when ordering assemblies from Renton Lot Time

Split Order Timing Threshold (days until SOS)

how many days prior to a SOS condition will an expediter wait until deciding to split an order

Supplier Reorder Lead Times (days)

the lead time required when ordering components from outside suppliers

Supplier Order Quantity (days)

the order quantity (determined by time span coverage) used when ordering from outside suppliers

Purchased Outside Production Schedule Advance (days)

this is a time buffer, prior to the need date, used when ordering from external suppliers

Table B-3.4 - Parameter & Variable Values & Sources

Description	Source	Value
CC 352 Production Rate	757 Master Schedule	2.5 - 4 day rate
Assembly Order Qty	Boeing Doc: M12-05	42 days
Assembly Flowtime	Boeing Doc: M12-02	30 days
Assembly Rate of Use	Assembly Part Record	1.0
Assembly S & C	Assembly Part Record	0
Assembly Processing	Historical Data & Model Testing	30 days
Stringer Processing	Historical Data & Model Testing	36 days
Stringer Flowtime	Boeing Doc: M12-02	71 days
Stringer Order Qty	Boeing Doc: M12-05	35-200 units
Stringer Rate of Use	Stringer Part Record	1.0
Stringer S&C	Stringer Part Record	0
Doubler Processing	Historical Data & Model Testing	40 days
Doubler Flowtime	Boeing Doc: M12-02	83 days
Doubler Rate of Use	Doubler Part Record	1.0-1.5
Doubler S&C	Doubler Part Record	0-9 units
Web Processing	Historical Data & Model Testing	175 days
Web ROLT	Boeing Doc: M12-05	180 days
Web Order Qty	Boeing Doc: M12-05	252 days worth
Web Schedule Advance	Boeing Doc: M12-05	42 days

Description	Source	Value
Clip Processing	Historical Data & Model Testing	185 days
Clip ROLT	Boeing Doc: M12-05	180 days
Clip Order Qty	Boeing Doc: M12-05	252 days worth
Clip Schedule Advance	Boeing Doc: M12-05	42 days
Rejection Rates	Estimated	0-100 Percent of Order
Order Priorities	Boeing Doc: M4-04	AOG, SOS, Pri 1, Pri 7
Split Order Quantities	Estimated	25 % of Order Qty
SOS Time Fraction	Estimated	80 % of Normal Flow
Priority 1 Time Fraction	Estimated	90 % of Normal Flow
Split Order Threshold	Estimated	10 days until SOS
Split Order Flowtimes	Estimated	5 days