

**SIMULATION FOR DECISION SUPPORT
IN A VIDEO CASSETTE
PRODUCTION PLANT**



by
Gökalp YILDIZ

August, 1997
İZMİR

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

SIMULATION FOR DECISION SUPPORT IN A VIDEO CASSETTE PRODUCTION PLANT

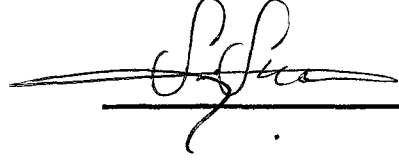
**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylül University
In Partial Fulfillment of the Requirements for
the Degree of Master of Science in Industrial Engineering,
Industrial Engineering Program**

**by
Gökalp YILDIZ**

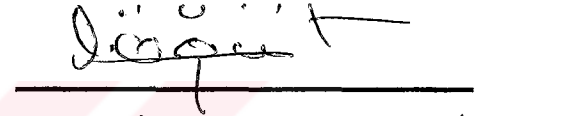
**August, 1997
İZMİR**

M.Sc THESIS EXAMINATION RESULT FORM

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Assist. Prof. Dr. Semra Tunali
(Advisor)

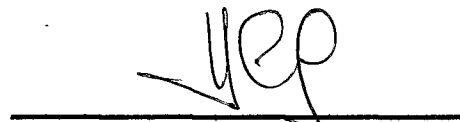


Prof. Dr. Cevdet ÖGÜT
(Committee Member)



Prof. Dr. Kutluhan Y. Zoral
(Committee Member)

Approved by the
Graduate School of Natural and Applied Sciences



Prof. Dr. Macit Toksoy
Director

ACKNOWLEDGEMENTS

I would like to give special thanks to my advisor Assist. Prof. Dr. Semra Tunalı who spent many hours reading early drafts of this thesis and made numerous improvements to the manuscript.

I wish to give special thanks to Industrial Engineer Serkan Büyükkaan from Raks Electronics Co. for his constructive suggestions along my plant trips. I wish to thank also my colleagues, Ayfer Avşar and Özcan Kılınçcı for their understandings during the writing of this thesis, Şeyda Topaloğlu, İlkey Saraçoğlu and Zuhall Serdar for their encouragements. Finally, I would like to thank my family for their great supports, encouragements, and helps.

Gökalp Yıldız

ABSTRACT

This thesis deals with the use of simulation as a decision support tool in a video cassette production plant which consists of four main departments. Each department consists of highly automated production lines. The objective of this thesis is to evaluate the performance of alternative work force allocation strategies in meeting the predefined production target for each department under given capacity constraints. The simulation models which serve to evaluate the alternative operating strategies has been developed in ARENA V2.2.



ÖZET

Bu yüksek lisans tezi, benzetim tekniğinin, video kaset üreten bir fabrikada karar destek aracı olarak kullanılmasıyla ilgilidir. Dört ana bölümden oluşan bu fabrikanın her bir bölümü, ileri üretim teknolojilerine sahip üretim hatlarından oluşmaktadır. Bu tezin amacı, önceden belirlenmiş bir üretim hedefini gerçekleştirebilecek olan alternatif işçi kullanım stratejilerini, her bir bölüm için, verilen kapasite koşulları altında değerlendirmektir. Alternatif işletme stratejilerini değerlendirmek için kullanılan benzetim modelleri ARENA V2.2' de geliştirilmiştir.



CONTENTS

	Page
Contents	VII
List of Figures	X
List of Tables	XI

Chapter One

INTRODUCTION

1. Introduction	1
1.1 Simulation As A Decision Support From Design To On-Line Production Scheduling	2
1.1.1 Simulation	2
1.1.2 Decision Levels In Manufacturing Organizations	4
1.1.3 The Use Of Simulation As A Decision Support Tool	5
1.2 Research Objectives	7
1.3 Application Environment	8
1.4 Recent Research Areas In Simulation	8
1.4.1 Expert Systems And Simulation	8
1.4.2 Object-Oriented Technology And Simulation	11
1.4.3 Parallel And Distributed Simulation	13

Chapter Two

MODELING AND SIMULATION

2. Overview Of Modeling And Simulation	14
2.1 Modeling And Simulation Approaches	15
2.1.1 Modeling Approaches	15

2.1.2 Simulation Approaches	17
2.2 Required Steps For Successful Simulation Modeling	19
2.2.1 Analyzing The Problem And The System	20
2.2.2 Selecting Input Probability Distributions	21
2.2.3 Building The Simulation Model	24
2.2.4 Verifying The Simulation Model	26
2.2.5 Validating The Simulation Model	28
2.2.6 Design Of Simulation Experiments	31
2.2.7 Statistical Analysis Of Outputs	32
2.2.8 Variance Reduction Techniques	33
2.3 Software For Simulation	36
2.3.1 General Purpose Languages	37
2.3.2 Simulation Languages	38
2.3.3 Simulators	40

Chapter Three

SIMULATION FOR DECISION SUPPORT IN A VIDEO CASSETTE PRODUCTION PLANT

3. Simulation For Decision Support In A Video Cassette Production Plant . .	41
3.1 Problem Definition	41
3.2 Overview Of The System	43
3.2.1 The Products	45
3.2.2 The Assembly Department	47
3.2.3 The Loading Department	52
3.2.4 The Packaging Department	53
3.3 Building The Model Of The Assembly Department	55
3.3.1 Assumptions Of The Model	57
3.3.2 Input Data Of The Model	58
3.3.3 Definition Of The Attributes, Variables, And Elements	82
3.3.4 Verifying The Model	87
3.3.5 Validating The Model Of An Assembly Line	88

3.3.6 The Results Of Experimental Studies For The Assembly Department	90
3.4 Building The Model Of The Loading Department	94
3.4.1 Assumptions Of The Model	95
3.4.2 Input Data Of The Model	96
3.4.3 Definition Of The Attributes, Variables, And Elements	98
3.4.4 Validating The Model Of A Loading Line	101
3.4.5 The Results Of Experimental Studies For The Loading Department.	103
3.5 Building The Model Of The Packaging Department	105
3.5.1 Assumptions Of The Model	107
3.5.2 Input Data Of The Model	107
3.5.3 Definition Of The Attributes, Variables, And Elements	111
3.5.4 Validating The Model Of A Packaging Line	113
3.5.5 The Results Of Experimental Studies For The Packaging Department.	114

Chapter Four CONCLUSIONS

4. Conclusions	121
References	125
Appendices	130
Appendix A.1	130
Appendix A.2	138
Appendix B.1	142
Appendix B.2	146
Appendix C.1	151
Appendix C.2	156

LIST OF FIGURES

	Page
Figure 3.1 The General Layout Of The Departments	43
Figure 3.2 The Bills Of Material Of A V0	44
Figure 3.3 The General Work Flow Between The Departments	45
Figure 3.4 The Layout Of The Assembly Lines In The Assembly Department	47
Figure 3.5 The Overview Of An Assembly Line	48
Figure 3.6 The Work Flow Between The Stations In An Assembly Line ...	50
Figure 3.7 The Overview Of The Loading Department	52
Figure 3.8 The Overview Of The Packaging Department	54
Figure 3.9 The General Algorithm For A Main Assembly Station	87
Figure 3.10 The General Algorithm For A Loading Line	101
Figure 3.11 The General Algorithm For A Packaging Line	113

LIST OF TABLES

	Page
Table 1.1 Information Requirements By Decision Levels	5
Table 3.1 The Four Products Having The Highest Production Volume	47
Table 3.2 The List For The Operations Of An Assembly Line	49
Table 3.3 The Automated Operations In A Packaging Line	54
Table 3.4 The Cycle Times Of The Main Assembly Stations	60
Table 3.5 The Batch Sizes Of The Components	60
Table 3.6 The Number Of Components To Be Used In Each Assembly Machine	60
Table 3.7 The Warning Levels Of The Components	61
Table 3.8 The Distances Between The Warehouse And The Main Assembly Stations	62
Table 3.9 The Distances Between The Main Assembly Stations	62
Table 3.10 The Lengths Of The Conveyor Segments Of An Assembly Line	63
Table 3.11 The Time Between Failures In An Assembly Line	64
Table 3.12 Curve Fit Summary For The Operation Of Feeding Tape Pad In The Assembly Department	65
Table 3.13 Curve Fit Summary For The Operation Of Feeding Guide Roller In The Assembly Department	66
Table 3.14 Curve Fit Summary For The Operation Of Feeding Guide Pin In The Assembly Department	67
Table 3.15 Curve Fit Summary For The Operation Of Feeding Lock Plate In The Assembly Department	68
Table 3.16 Curve Fit Summary For The Operation Of Feeding Lock Plate Spring In The Assembly Department	69
Table 3.17 Curve Fit Summary For The Operation Of Feeding Release Lever In The Assembly Department	70

Table 3.18	Curve Fit Summary For The Operation Of Feeding Reel Lock In The Assembly Department	71
Table 3.19	Curve Fit Summary For The Operation Of Feeding Reel Lock Spring In The Assembly Department	72
Table 3.20	Curve Fit Summary For The Operation Of Feeding Tape Guide Pole In The Assembly Department	73
Table 3.21	Curve Fit Summary For The Operation Of Feeding Pivot In The Assembly Department.....	74
Table 3.22	Curve Fit Summary For The Operation Of Feeding Reel Flange In The Assembly Department	75
Table 3.23	Curve Fit Summary For The Operation Of Feeding Leader Tape In The Assembly Department	76
Table 3.24	Curve Fit Summary For The Operation Of Feeding Tape Clamp In The Assembly Department.....	77
Table 3.25	Curve Fit Summary For The Operation Of Feeding Clear Window In The Assembly Department.....	78
Table 3.26	Curve Fit Summary For The Operation Of Feeding Reel Spring In The Assembly Department.....	79
Table 3.27	Curve Fit Summary For The Operation Of Feeding Front Cover In The Assembly Department.....	80
Table 3.28	The Input Models Used For Feeding Operations	81
Table 3.29	The Simulation Results For The First Assembly Line	89
Table 3.30	The Factor Plan For The Assembly Department	90
Table 3.31	The Replication Conditions For The Assembly Department	91
Table 3.32	Throughputs Of The First Assembly Line For Each Factor Level....	91
Table 3.33	The Results Of The First Policy For The Assembly Department.....	92
Table 3.34	The Results Of The Second Policy For The Assembly Department ..	92
Table 3.35	The Results Of The Third Policy For The Assembly Department....	93
Table 3.36	95% Confidence Intervals For The Performance Values	93
Table 3.37	The Distances Between The Pancake Buffer And The Loading Machines.....	98
Table 3.38	Curve Fit Summary For The Operation Of Feeding Pancake In The Loading Department.....	99

Table 3.39	Simulation Results Of Ten Replications For A Loading Line	102
Table 3.40	The Factor Plan For The Loading Department.	103
Table 3.41	The Replication Conditions For The Loading Department	104
Table 3.42	The Summary Results For The Loading Department	104
Table 3.43	The Operation Times Of The Packaging Machines	108
Table 3.44	Curve Fit Summary For The Operation Of Casing In The Packaging Department	109
Table 3.45	Curve Fit Summary For The Operation Of Large Casing In The Packaging Department	110
Table 3.46	The Simulation Results Of Ten Replications For The First Packaging Line	114
Table 3.47	The Factor Plan For The Packaging Department	115
Table 3.48	The Replication Conditions For The First Stage Of Experiment	116
Table 3.49	Summary Results For The First Policy In The Packaging Department	116
Table 3.50	Summary Results For The Second Policy In The Packaging Department	117
Table 3.51	The Replication Conditions For The Second Stage Of Experiment . . .	118
Table 3.52	The Summary Results For The Packaging Department	119

CHAPTER ONE

INTRODUCTION

1. Introduction

Nowadays, manufacturing organizations spend tremendous effort to stay competitive in constantly changing global markets. The products which are not adequate to the customer specifications in quality and in price do not have any chance to compete with other products in global markets. Hence, the manufacturing organizations need to be dynamic in order to have the capability to meet the changing needs of the market. This dynamic nature of manufacturing organizations requires the managers to make timely and accurate decisions under the uncertainty. To efficiently cope with the changing needs of the market, the managers need to use the right decision support tool at the right time. One of the new generation decision support tools is simulation. Although, simulation has been around for decades, the power of simulation was realized only when the advanced PCs and sophisticated simulation tools became widespread (Banks&Norman, 1995).

In general, the decision makers use the models which represent the logical and the mathematical relationships between the components of the systems to analyze the manufacturing systems. Models of manufacturing systems can be put into two main categories: Optimization and performance models. Optimization models generate a single or a set of decisions for a given set of criteria and constraints. These models use the principles of optimization. Performance models, on the other hand, estimate the measures of system performance for a given set of decisions and system parameters. Performance models use the techniques of stochastic processes, probability, and computer simulation. The recent developments in computer simulation have been causing an increase in the use of simulation for studying a variety of operational policies and their effects on the system performance (Altiook,

1997, pp.1-6).

1.1 Simulation As A Decision Support Tool From Design To On-Line Production Scheduling

Historically, simulation has been used primarily as a design tool for new projects or major enterprises, such as establishing a complex production system. This is the traditional use of simulation in a manufacturing environment. However, the only one time use of a simulation model which usually requires a long lead time to develop is not feasible. Therefore, the use of simulation extends to the new areas in manufacturing organizations. These are the use of simulation as a planning and controlling tool. Since the use of simulation as a controlling tool is viable for highly advanced manufacturing systems, it is a very new approach and it is not widely utilized in commercial applications yet (Pritsker, 1992).

1.1.1 Simulation

The simulation is an operations research based technique. It can be defined as the process of building the model of a system, conducting experiments on the model and analyzing the responses of that model of the system. Simulation is not the only operations research technique which is available to analyze the manufacturing systems. A lot of analytical methods have been developed in this area. However, these analytical methods are not applicable to advanced manufacturing systems. It is difficult to represent complex interactions between the system components with a set of constraints. The power of simulation is originated from its success in analyzing complex systems. Simulation has generally been applied in problem areas where the variables and their relationships are clearly understood but where no efficient analytical solution method exists.

A simulation model shows the essential features of the system to be analyzed. The details of the simulation model depend upon the objectives of the simulation study. By using the accurate input data, the response of the system can be observed and the

information which was obtained can be used to support a decision making process. The cliché ‘Garbage-in, Garbage-out’ emphasizes the importance of the input data to be used in a simulation model.

Since the simulation is a process, it consists of several steps. The first step in a simulation study is to define the problem. Defining the problem is one of the most important steps in a simulation study. It must be noted that defining the problem limits the level of detail to be included in a simulation study. The second step is to understand the system. It is possible to gain lots of information about the system by observations, interviews and the CAD drawings of the facilities. The third step is the selection of the performance measures. The performance of alternative decision policies is compared according to these measures. The fourth step of a simulation study requires determining what data are needed and what data are available. At this step, the simulation modeler needs to have good skills in statistics to be able to analyze these data. The fifth step is to develop assumptions about the problem. The assumptions need to be sufficient to be able to answer the questions which are asked. The sixth step involves the model building. The model represents the mathematical and the logical relations between the resources of the system. The seventh and the eighth steps are the selecting the software to be used and the codifying the model by using the selected software, respectively. The ninth and the tenth steps cover the model verification and the model validation. The model verification refers to the debugging of the model. At this step, syntax and logical errors are eliminated to make sure that the code runs as it is intended by the model builder. On the other hand, the model validation guarantees that the simulation model which was developed represents the real system. The last steps are the designing of experiments, running the simulation model according to pre-selected design points and analyzing the output of the simulation model. Experimental design involves determining the factors which affect the model response, specifying the factor levels and conducting simulation runs. Statistical analysis of the output is the one of the most important steps in a simulation study. This step helps to convince the model builder about the reliability of the simulation output.

Two major types of simulation software for manufacturing applications are available. These are simulation languages and simulators. Simulation languages are described as having essentially unlimited modeling flexibility, but they require programming expertise. Simulators, on the other hand, are often much easier to use, but they are not adequate and flexible to model the complex systems. The general purpose computer languages can also be used to simulate a system. However, if the system to be simulated is a very complex one, it is very difficult to simulate the system by using the general purpose computer languages such as C, FORTRAN, and PASCAL.

1.1.2 Decision Levels In Manufacturing Organizations

All of the managerial functions such as planning, organizing, leading and controlling are necessary for successful organizational performance. No one function is more important than another. The problem in any managerial function prevents the success of the manufacturing organizations. Therefore, establishing an effective control system is very important. If managers do not know where their organizations are leading for and do not make appropriate corrections and adjustments as they progress, all their managerial functions may be affected. To establish an effective control system, the decision makers should quickly and accurately receive the information which is required for decision process. In a Computer Integrated Manufacturing (CIM) environment, Management Information System (MIS) provides the right information at the right time. The MIS of manufacturing organizations generally consists of a series of information systems of varying degrees of complexity and scope (James&Stoner, 1982, pp.641-661).

Three levels of decisions are made in manufacturing organizations. These are strategical, tactical and operational decisions. The overall goals of the manufacturing organizations are established at the strategic level. The overall goals of the manufacturing organizations may concern with the product and production system development, long term planning, and very costly renewal investments. Production is planned at the tactical level. At this level, production plans, process plans, and

schedules are prepared. In a CIM environment, MRP, MRPII are used for planning all kinds of inputs such as labor, inventory, and demand. Operational decisions are made at the lowest level. These types of decisions require highly detailed and accurate information on a daily or weekly basis. Table 1.1 represents the information requirements by decision levels (James et al., 1982, p.647).

Table 1.1 Information Requirements By Decision Levels

Characteristics of Information	Operational Decisions	Tactical Decisions	Strategical Decisions
Source	Largely Internal	↔	External
Scope	Narrow	↔	Very Wide
Level of Aggregation	Detailed	↔	Aggregate
Time Horizon	Historical	↔	Future
Currency	Highly Current	↔	Quite Old
Required Accuracy	High	↔	Low
Frequency of use	Very Frequent	↔	Infrequent

As it is shown in Table 1.1, the scope of information is narrow in the operational level, but very wide in the strategical level.

1.1.3 The Use Of Simulation As A Decision Support Tool

As it is explained above, there are three levels of decisions in manufacturing organizations. Simulation can be used in all of these levels to support the decision maker (Thompson, 1994). At the strategic decision level, simulation is used for providing information to decision makers on how the system will operate under a given design configuration. Simulation has been widely used in the design of the advanced manufacturing systems. For example, simulation is extremely valuable at the detailed design stage of Flexible Manufacturing Systems (FMS), when the overall structure of the FMS needs to be determined. Simulation is also used for figuring out the feasible design of conventional manufacturing systems. Although simulation is a powerful design tool, the one time use of simulation is not feasible because of the long lead time to build a simulation model (i.e., collecting input data, model building, verifying, validating, output analyzing). The following design questions can be answered by using simulation as a design tool:

- How many machines are required to meet the annual demand?
- What is the maximum length of conveyor?
- How many AGVs are required?
- What is the minimum number of operators to keep the automated line operating?

The use of simulation at the tactical decision level involves evaluating the operating policies of a particular manufacturing system. Simulation is used as a planning tool at this level. When the simulation is used for the purpose of scheduling or detailed planning, production data is extracted from the central database. The simulation model of that manufacturing system uses these production data. Before the actual production, the manufacturing system is simulated and the response of the model is evaluated. Many problems in the production schedule or plan can be seen before the operation of real system. For example, if the delivery times can not be satisfied for some jobs, an alternative production schedule is experimented on the simulation model (Tunalı, 1997, pp.243-250).

Since the 'on-line' issues are applicable only for advanced manufacturing systems, the use of simulation at operational level is a very new approach. At the operational level, the simulation is used for making decisions and controlling. For example, simulation is used to produce the schedules which can be distributed to the shop floor for describing the sequence for jobs which are to be processed. There are several reasons for the advent of the simulation based on-line scheduling systems (Coskun&Oren, 1994, p.293):

- Conventional production scheduling models are not sufficient to represent a realistic factory model. They usually underestimate the complexity of a manufacturing environment.
- Realistic scheduling problems can not be solved by algorithmic methods.
- Conventional scheduling techniques cannot manipulate unexpected events in a dynamic manufacturing environment.

If an unexpected event occurs, the available schedule may be invalidated. In this case, a fast and reliable generator for schedule is highly desirable. The simulation based on-line scheduling systems can solve these problems by generating new schedules based on recent system status (real-time data). However, it is very difficult to establish an automated data collection system to monitor all of the system variables which represent the current status of machines, labors, and queues (Norman, 1992).

1.2 Research Objectives

This thesis aims at using the simulation as a decision support tool in a video cassette production plant. The plant consists of four departments. These are the plastic injection, assembly, loading and the packaging departments. The simulation model includes assembly, loading and packaging departments except for the plastic injection department. The model is intended to be used as a decision support in evaluating the following operational issues:

Assembly Department;

- How many transporters (workers with hand trucks) should be utilized in the assembly department to keep the assembly lines operating?
- How many technicians should be utilized per assembly line in the assembly department to meet the desired production quantity?
- How many component feeding operators should be utilized for each assembly line to meet the desired production quantity?

Loading Department;

- How many pancake feeding operators should be utilized for each loading line to meet the desired production quantity?

Packaging Department;

- How many casing operators should be utilized per packaging line to meet the desired production quantity?

- How many large casing operators should be utilized per packaging line to meet the desired production quantity?

1.3 Application Environment

The video cassette manufacturing plant was selected for application environment. The plant consists of four departments. These are plastic injection, assembly, loading and packaging departments. The selected product which is called VHS has the highest production volume annually (%96 of total production). All of the departments consist of highly automated machines, conveyors, and workers.

1.4 Recent Research Areas In Simulation

In recent years, many researches in the field of simulation have tried to enhance the power of simulation by integrating it with other engineering tools such as artificial intelligence and analytical methods. Artificial intelligence is the most important field which supports the simulation process in many ways. Although, these two fields developed independently, there are many artificial intelligence concepts which have found actual and potential use in a simulation environment. The best example for the artificial intelligence concepts is expert systems. Expert systems are widely used in various phases of a simulation study such as model building, model validating, input and output analyzing, etc. The object oriented technology is the other example which makes the modeling environment highly flexible.

1.4.1 Expert Systems And Simulation

O' Keefe has proposed taxonomy for the combination of expert systems and simulation models. He considers four kinds of design choices. The first two involves the embedding of one within the other and vice versa. The third one considers simulation systems and expert systems which are designed, developed and implemented as separate software systems but they may interact. The last one suggests both systems to share some data and cooperate in processing of a task (O' Keefe,

1986, pp.10-15). All of these combinations can provide useful results, especially, when expert systems are used to help the simulation modeler in developing and using simulation. This includes decision making within a simulation, model generating, data analyzing, model validation and verification.

There are many studies about the use of simulation as a decision support tool. In general, simulation models are used for “what-if” analysis in various domains such as manufacturing control, business, health-care, quality assurance, etc. In these cases, there is a need for a software system which will enable the user to explain his problem in the right way, to feed it to the model and then to report the adequate message to the user. Simulation based on-line scheduling of FMSs exemplifies this category. This kind of integration is called as knowledge based simulation. Such an integration results in realistic factory modeling, unexpected event handling, inclusion of human expertise on decision making and obtaining improved production schedules in real time. In general, simulation based on-line scheduling systems consist of four submodules. The first one is simulation submodule. This module evaluates the performance of the alternative schedules based on current system status. The second one is the expert submodule which takes the place of a human expert. The expert submodule selects the appropriate schedule with respect to a set of selection criteria defined in the knowledge base. The third one is the data manipulation module. It provides the adequacy between the input and the output data of the simulation and the expert modules. The last one is the user interface which provides communication between the user of the system and the system. The studies of Manivannan, Banks and Kusiak are the milestones in the area of knowledge-based simulation (Kusiak&Chen, 1988, pp.113-130).

As it was discussed before, a simulation study requires a long lead time to develop the model and to carry out experiments on the model. This seriously limits its utilization. Usually, a simulation modeling study requires the analyst to have a background in statistical methods and simulation languages. It takes training and practice before an analyst can perform a moderately complicated simulation study with a commercially available simulation language. Furthermore, even for an

experienced analyst, modeling and programming of a complicated system can be a very time consuming process. The advantages of simulation can not be fully exploited unless the time consuming phase of learning and using a simulation language is reduced. Simulation model generators are designed to speed up the modeling and programming stage of simulation. These kinds of generators can generate the simulation code automatically. Most of the expert systems developed so far aim at automating the development of the code. For example Haddock proposed a simulation generator for flexible manufacturing systems design and control. This generator consists of subroutines which were written, compiled and linked with SIMAN (Haddock, 1988, pp.22-31).

Expert systems are widely used in analyzing the input data and the output of the simulation model. Simulation is not an optimization tool. Executing the simulation model once to obtain an answer is not possible because simulation is only a statistical sampling technique. The modeler makes multiple runs, adjusts input parameters between the simulation runs to produce the results that might be judged better according to specified criteria. Accordingly, after the completion of a simulation run, the modeler studies the simulation outputs which characterize the run, then tries to identify the factors which limit the system's observed performance and makes system adjustments that might improve performance. SESSA which was developed by Mellichamp and Park is a good example for input and output data analyzing. SESSA supports the simulation analyst from beginning to end of a simulation study. All of the simulation steps require statistical issues. SESSA includes the following submodules (Mellichamp&Park, 1989, pp.134-139):

- Collecting data to estimate values for input variables.
- Estimating values for input variables
- Identifying standard distributions for input variables.
- Comparing model response estimates to system estimate
- Assessing the impact of the initial conditions.
- Determining run lengths and number of replications.

- Estimating response variable values from output.
- Constructing confidence intervals for estimates.
- Associating standard distributions with response variables.
- Comparing output from alternative designs.

As it is explained above, SESSA is an interactive expert statistical consultant that provides assistance in simulation steps which involve statistical issues.

The other important research area in simulation is Object Oriented Technology. This approach makes the simulation model more flexible and modular.

1.4.2 Object-Oriented Technology And Simulation

Object-Oriented Technology (OOT) is based on the artificial intelligence concepts. OOT is used for knowledge representation within a simulation model. The entities of a discrete event simulation can be considered as objects in an object oriented formulation. Two types of knowledge are associated with each entity. These are factual and behavioural knowledge. An object's factual data can be viewed as the attributes of the entity, characterizing its capabilities, current state or parameters. The objects behavioral data are essentially procedural functions that characterize how the entity is to behave in certain circumstances. Messages which are sent to an object result either in the provision of selected factual data or the initiation of a specific behavior. Object oriented technology provides a stable structure by which the interactions between the components of the modeling system can be specified. It also ensures that a user can make incremental changes to selected parts of the modeling system without threatening the integrity of the entire system.

Frames provide the structure that supports object oriented technology. Each frame contains slots. An object's factual data items can be placed in these slots. The power of frame based knowledge representation arises from the inheritance capabilities, the manner in which the frames can be related to one another. The frame concepts fit very naturally into the structure of discrete event simulation, offering the modeler a number

of capabilities. These capabilities can be as follows (Nielsen, 1991, pp.1-19):

- **Library development:** New entities can be specified as members of a particular class with certain differences. These differences may relate to the addition or subtraction of slots, to the modification of slot values, or to the adjustment of relationships. New entities, whether actual or class representations, can be specified much more efficiently in this way.
- **Construction via component assembly:** Systems can be assembled by selecting components from a library then connecting those entities with various relations. Each entity is thus created fully formed, complete with data, relations and necessary procedures.
- **Component representations:** Component representations can be used efficiently to manipulate subcomponents. For example, if it is assumed that there are three frames representing a machine, an input buffer and output tray, the subcomponents can be linked to the machine by “has-a” and “belongs-to” relationships. Then if an instance of machine is created at run time, the associated input buffer and output tray instances will also be created and properly linked. Consequently, the frame relationships can provide the modeler with considerable power.

Object-oriented technology also brings the graphical based editing. An icon, menu or other visual representation can be associated with each frame. Positioning the cursor on an appropriate icon or menu will send a message to the associated object, resulting in an appropriate action being taken. Graphical based editing provides user friendly interfaces, rapid prototyping of the simulation model, less error and less training requirements. Additional information about the object-oriented applications in the simulation can be seen in the studies of Lefrancois and Montreuil, Ouzrout and Senoune, Corey and Clymer.

1.4.3 Parallel And Distributed Simulation

The concept of parallel and distributed simulation refers to the execution of a simulation model on a machine containing multiple central processing units (CPU). This includes execution on closely coupled multiprocessor machines (parallel simulation) as well as geographically distributed computers (distributed simulation). The field has been studied over the last 15 years (Fujimoto, 1993, pp.106-113). However, the deployment of the parallel and distributed simulation techniques to the general simulation community has been rather slow. One of the reasons has been the unavailability of the general purpose hardware which allows the parallelization of simulation applications (Salmi&Harju&Porras, 1994, pp.153-157).

As computers become more powerful and their use expands, the need to simulate larger and more complex systems in reasonable computing times becomes more important. For example, simulations of only modest sized telecommunication networks may require weeks on a modern workstation. Parallel simulation can significantly speedup the process. Therefore, it can be said that the need for parallel simulation is originated from the need for less execution time of the large and complex models. In addition to the less execution time, some simulation models may require geographically distributed resources, e.g., humans, or databases which are costly and difficult to move. In these kinds of cases, it is inevitable to use the technique of distributed simulation.

CHAPTER TWO

MODELING AND SIMULATION

2. Overview Of Modeling And Simulation

Manufacturing in today's global marketplace is becoming very complex. In 1990s, the factors like high quality, quick delivery, low cost and responsiveness to the changing demand are considered as key to successful competition of the manufacturing organizations.

Industrial engineers nowadays luckily have many new tools to select from in order to successfully cope with the changing needs of the market. Most of these new tools are based on the earlier versions of the old tools. The improvement in the software and hardware technology coupled with the urgent need of manufacturing organizations for a powerful tool makes these old tools, such as simulation, more important and more useful. Although, simulation has been around for decades, the importance of simulation was realised only when the advanced PCs and software were brought to the market.

Simulation is defined as the process of building a model of a real system and conducting experiments on this model for the purpose of either understanding the behaviour of the system and/or evaluating various strategies for the operation of the system (Shannon, 1992, p.65). The most important terms in this definition are the system and the model. System can be defined as a collection of entities that act and interact together toward the accomplishment of some predefined goals (Law&Kelton, 1991, p.3). To analyse a system, a model of the real system needs to be established. This can be done by a set of assumptions, which usually take the form of mathematical or logical relationships. The model which is established under these sets of assumptions helps to gain some understanding of how the corresponding system

behaves. Accordingly, simulation model can be considered as a mathematical model that is studied by means of simulation.

Simulation is one of the most important and most commonly used operations research technique in 1990s manufacturing environment. The factors such as flexibility and being powerful in representing the complex relationships between the objects can be attributed to the widespread use of simulation. Manufacturing environment is not the only area where simulation is used (Kiran&Kaplan&Unal, 1993). Simulation is also widely used in the areas of computer systems, business, government, ecology, environment, and bioscience (Shannon, 1992, p.65). This thesis particularly deals with the use of simulation in a manufacturing environment.

2.1 Modeling And Simulation Approaches

When the modeling and simulation process start, the decision on how the system can be modeled and simulated needs to be made. The statement “modeling and simulation approaches” refers to the alternative methods or logic used in modeling and simulation.

2.1.1 Modeling Approaches

A system can be studied by two ways. The first approach is to carry out a set of experiments on a real system. However, experimenting on a real system may not be possible due to two reasons. The first reason is that there may not be a real system on which a set of experiments will be conducted, such a case appears in designing phase of a system. The second reason is that conducting a set of experiments on a real system may take a long time and results in high cost. For example, manufacturing systems can not be stopped for comparing the performance of alternative operational strategies on existing production resources because the opportunity cost of this action will be very high.

The second approach is to experiment on the model of the real system. The intent of a model is to represent the relationships between the objects in the system. The advantages like requiring less time and money and also not disturbing the behaviour of the real system cause the modeling approach to be widely used in analysing manufacturing systems. The model of the real system can be two types. The first one is a physical model of an actual system. These kinds of models are called as iconic models. These are usually used for training purposes. The second one is a mathematical model of a real system. A mathematical model represents a system in terms of logical and quantitative relationships. After a valid mathematical model of a real system is established, it is possible to change and manipulate these logical and quantitative relationships to see how the model responds to these changes.

The model can be studied either by means of analytical methods or simulation. The choice depends on the complexity of the model. If the model is not a very complex one, the choice needs to be an analytical method. Although, analytical methods guarantee an optimum solution, they often are used under simplified assumptions. If the model is a very complex one such as the model of an advanced manufacturing system, the choice needs to be simulation because the complex interactions between the components of the advanced manufacturing systems are difficult to represent by means of a constraint-based set of equations which are necessary to reach an analytical solution. On the other hand, simulation simply produces reports on some predefined system performance measures. It does not provide optimum solutions. Conducting a successful simulation study requires the simulation modeler to be competent on computers, simulation modeling, simulation languages and statistics.

This thesis aims at studying an advanced manufacturing system which consists of automated assembly machines, a conveyor system, hand trucks and workers. The simulation technique has been chosen as a tool to study the complex interactions between the components of this dynamic and stochastic system.

2.1.2 Simulation Approaches

A simulation project aims at collecting statistics about a particular system as a function of time. From that standpoint, there are two distinct types of simulation. The first one is discrete simulation. In discrete simulation, statistics are collected only at selected points in time when certain changes take place in the system. These selected points represent events which are defined as instantaneous occurrences that may change the state of the system. Accordingly, the events do not take place in time. The event that has been defined can be either conditional or unconditional. An unconditional event entirely depends on scheduled time. However, a conditional event may involve other conditions such as the state of the system components in addition to scheduled time (Pegden&Shannon&Sadowski, 1990, p.24). The second one is continuous simulation. Continuous simulation concerns with the system in which the state variables change continuously with respect to time. Continuous simulation has important applications such as calculating the average tank level in a fuel-oil production system. Typically, continuous simulation models involve differential equations that give relationship for the rates of change of the state variables with time (Law et al., 1991, p.109). In addition to these two simulation types, another simulation type called as combined simulation appears in many publications in simulation. However, there is no need to classify combined simulation as a third type, because it is only a combination of discrete and continuous simulation.

A simulation model which is discrete or continuous depending on the state variables of the system needs to be codified in an appropriate computer language. General purpose computer languages (C, PASCAL, FORTRAN, etc.), and simulation languages (SIMAN, ARENA SE, SIMFACTORY, GPSS, PROMODEL, etc.) can be used to codify the model which represents a system. Since the vast amount of problems is associated with discrete systems, there are quite many simulation languages having special features to model the discrete systems. The use of events is fundamental to the development of all discrete simulation languages. These discrete simulation languages also have the capability of simulating continuous systems.

Three simulation approaches are used to simulate systems. These are event scheduling, activity scanning, and process interaction. In each of these approaches, the concept of event is used to advance the simulation clock. When the next event is selected for processing, the corresponding model logic is executed to make appropriate changes in the model state (Pegden et al., 1990, p.24).

The event scheduling approach involves a succession of unconditional events over time. The system to be modeled is described in terms of status disturbing events. The modeler establishes a simulation model by defining each event that can occur in the system, specifying the causes and effects of each event, creating mechanisms to execute event change, logically linking events to one another, providing for the update of time and statistics at each event, and collecting statistics of interest. The event scheduling time control procedure then selects the event having the earliest occurrence time from the event list, updates the simulation clock, and invokes the corresponding event routine. Any condition testing other than simulation clock must occur within event routines. This process stops when the termination event occurs. The termination event routine is executed when the simulation clock equals to predefined time or when the number of entities passed through the system equals to a predefined quantity. Event scheduling approach is used directly for codifying the model to be simulated by any general purpose computer languages. However, in the case of using simulation languages, event scheduling approach builds the fundamentals of simulation language logic, and the approach is used indirectly as an event routine that was written externally in general purpose computer language (Pegden et al., 1990, p.24).

The activity scanning approach chooses the next event based on both scheduled time and condition testing. The term activity is defined as a system state transition requiring a period of time. In some models, the events that are expected to occur can not be scheduled strictly on the basis of time. Each activity routine in a model specification describes all actions that must occur because the model assumes a particular state. This procedure is called activity scanning. In the activity scanning approach, the time control mechanism searches activities in priority order for

activation conditions. Then, the first activity routine whose activation conditions are met is executed. The search mechanism starts over again in priority order, and the process continues until the termination time (Pegden et al., 1990, p.25).

The process interaction approach views the world as a set of competing and cooperating entities modeled in terms of interacting concurrent processes. A process is a time-ordered sequence of interrelated events separated by passages of time, which describes the entire experience of an entity as it flows through a system. As in the case of event scheduling approach, the process interaction approach enables evaluating a simulation model as it evolves over time by executing the events in increasing order of their time of occurrence. Internally, these two approaches to simulation are very similar. They differ mainly in the language constructs (i.e., SEIZE, RELEASE, QUEUE, etc., blocks). Process interaction statements are more understandable in nature and automatically translate certain situations commonly occurring in a simulation model (Law et al., 1991, pp.237-240). When the process interaction approach is selected, the simulation approach which is actually executed behind the scene is the event scheduling logic. Codifying the simulation model from the process interaction point of view usually requires the use of special purpose simulation languages (Collins&Watson, 1993).

The simulation approach used in this thesis is process interactions. The process interaction world view requires the use of a simulation language. In this thesis, ARENA SE V2.2 has been used to model the application environment.

2.2 Required Steps For Successful Simulation Modeling

A simulation study consists of several steps such as problem definition, input data collection, software selection, model development, verification, validation, output analysis, etc. Successful implementation of these steps requires the model builder to have background in statistics and computers. However, some people think that a simulation study involves only software selection and coding. This is a common misconception for the new simulation analysts. If the steps given above are not

followed by the simulation analyst, it is impossible to obtain reasonable solutions to a specified problem (Law&McComas, 1992).

2.2.1 Analyzing The Problem And The System

Perhaps the most important, but often the most overlooked, aspect of simulation modeling is to accurately define the problem to be solved. The novice simulation modelers may think that when a simulation model is established, it will be able to provide answers to any and all questions relating to the simulated system. Like other computer applications, a simulation model can only do what it was designed to do. It is impractical to design the model to do everything. Therefore, careful determination of the model design (e.g. the scope and level of detail which is represented in the model) is critical. Constructing a list of specific questions that the model will answer is a good way to begin a simulation study (Banks&Gibson, 1996, p.34). At this stage, the decision on whether the simulation is a right tool for solving the problem on hand needs to be made. Although simulation is a powerful analysis tool, there are other tools available, perhaps less expensive or quicker to apply. These may include mathematical programming or spreadsheet analysis. From a practical standpoint, simulation models do not provide optimal solutions. They simply report on system performance for a set of conditions. To obtain an acceptable solution or answer, the user must carry out a number of simulation experiments and analyse the output.

After determining the problem to be solved and conforming that simulation is a right tool, the next step is to obtain accurate input data about the system to be simulated. The information about the system can be gained by observing the system, interviewing with the people who are responsible for the design of the system, talking with operators or supervisors and determining the CAD drawings of the system to be simulated. After gaining some understandings about the system, the system components and the operational characteristics of the system need to be listed. The system components might include machines, workers, conveyors, AGVs, robots, storage-queue locations and capacities, bottleneck points of the system, etc. The operational characteristics could be scheduling rules, processing logic, operating

schedules and breaks, etc. These lists can be summarized as a description of operations. Since the fundamentals of the simulation models are established at this step, it is one of the most important activity in a simulation study.

2.2.2 Selecting Input Probability Distributions

In addition to system operating policies (i.e., queuing discipline, buffer capacities, etc.), the simulation analyst will need a variety of input data including job interarrival time, processing time, the time between failures, repairing time, conveyor speed, the cycle time of an indexing table, etc. These input data may be either deterministic or stochastic. For example, the time to carry out manual operations and job interarrival times are stochastic and they usually have nature requiring the use of random number generators. However, the cycle time of an indexing table can be assumed constant since the process is fully automated. At this stage, the simulation analyst must decide what data are needed, what data are available, whether the data are pertinent, whether existing data are valid for the required purpose and how to gather the data. In most simulation studies, the determination of what data need to be used is a very difficult and time consuming task. The methods to follow in determining the input data depend on the existence of the system in question.

If the system exists, the actual data can be obtained from the historical data which are available or can be collected. In this case, the historical data which are collected or available can be used for obtaining simulation input models. Two methods can be applied. These are to sample directly from the empirical distribution or, to sample from a theoretical distribution if the data fit a theoretical distribution. Sampling from the empirical distribution makes the future same as the past, that is, it is assumed that the events can not be occurred, if they did not occur in the past. However, sampling from the theoretical distribution tends to make the data to have a large range. If empirical data are to be used, they are input in the form of a cumulative probability distribution. Observed values (or the values which are available) are put in the form of an empirical cumulative distribution by arranging them in ascending order, grouping identical values, computing their relative frequencies, and then computing their

cumulative probability distribution. If theoretical data are to be used, three steps are needed to be performed (Pegden et al., 1990, pp.44–46):

- The first step is to select a distribution which may represent the empirical data according to shape of the histogram.
- The second step is to estimate the parameters of the selected distribution by using the observed data.
- The last step is to determine the relative goodness of fit by using appropriate methods such as Chi-Square and Kolmogorov-Smirnov. Most of the dedicated simulation languages have some modules to perform these tasks automatically.

If the system is not yet exist then, there is no data about the past and no chance to gather any data. In this case (i.e., the system which is in design phase), it is impossible to obtain data. Therefore, other sources of input data are used. These can be data from similar systems, operator estimates and vendor claims. Since no system can show the same characteristics, using the data from a similar system may be risky. Operator estimates have wide range because humans are such poor estimators. They forget the extremes and emphasize the present. Vendor claims can be highly optimistic. For example, the reliability of the system can be grossly overestimated (Banks et al., 1996, pp.37-38).

In this thesis, the application environment also is an existing system where video cassettes are produced. In this environment, both stochastic and deterministic input data are needed to build a simulation model of the system. Deterministic input data are conveyor speed, worker speed, the cycle time of indexing tables, loading times and all of the operations which are carried by highly automated machines, etc. Stochastic input data are the time to machine failure, repairing times, the processing time of manual operations, etc. The historical data reports, available time study results, and the memory of automated machines are used as input data sources. To fit a theoretical distribution to the stochastic data, the input data analysis module of ARENA SE V2.2 was used. The details of input data analysis study will be explained in Chapter 3.

One of the requirements of any simulation model is the capability to generate random numbers. There are many algorithms available for generating the pseudo-random numbers. The term 'pseudo-random number' implies that each random number is generated from the preceding random number. This means that if the seed which is the starting value for generation algorithm is same for two simulation replications, same stochastic behaviour is obtained. A good random number generator needs to have the following features (Law et al., 1991, p.423):

- The generated random numbers need to be distributed $U(0,1)$ uniformly and they should not have any correlation with each other.
- The random number generator needs to be fast and to avoid the need for a large computer memory.
- The random numbers should be reproducible to enable a detailed analysis of alternative systems.
- The random number generator should have the capability to produce several separate random number streams.

The random number generation algorithms which are widely used are The Von Neumann's Midsquare method, The Linear Congruential Method, The Additive Congruential Method, and The Quadratic Congruence Method. These methods produce independent and uniformly distributed random numbers. In many simulation studies, it is not appropriate to use uniform distribution. The general approaches that take the uniform random number and transform it to the nonuniform distribution such as normal, exponential distributions can be classified as direct and indirect approaches. Direct approaches directly handle the desired distribution or random variable. Direct approaches are Inverse Transform Method, Composition Method and Convolution Method. If direct approaches prove to be inefficient, then the indirect approaches can be used. One of the most commonly used approach is Acceptance-Rejection Method (Houshyar&Nuila, 1993, p.54).

ARENA SE V2.2 has its own random number generator. The random number generator used in ARENA SE V2.2 is Linear Congruential Method. It has 10 random

number streams.

2.2.3 Building The Simulation Model

As it was mentioned before, the simulation modeling is defined as an approach reflecting the relationships between the actual system components by a computer without disturbing the behaviour of the actual system. Building a simulation model requires several steps. These can be summarized as follows:

- Selecting the performance measures which help the simulation analyst to compare the performance of alternative policies.
- Specifying the system components and the operational characteristics which directly affect the selected performance measures (making assumptions-limits the list for description of operations).
- Drawing algorithm schemes and flow charts which represent the entity flow and the control logic according to description of operations.
- Selecting a simulation tool (general purpose computer programming languages, special purpose simulation languages, or dedicated simulators) to codify these mathematical or logical relationships between the system components.
- Codifying the model in a selected simulation tool.

The effect of a decision or a design change on the behaviour of a system may be either measured by observations or estimated by using the methodologies such as simulation. This is usually called performance evaluation. Since the simulation is a performance evaluation technique, there is a need to select the criteria to be used to compare the performance of the alternative decision policies. Typical performance measures in a manufacturing environment are output rates or throughputs, average inventory levels, utilization, customer service levels, average flow times, and machine downtimes. In simulation studies, these performance measures need to be selected according to the problem to be solved. For example, if the problem is to determine the buffer locations and the capacities of each buffer in an automated flow line, the performance measure needs to be utilization of the whole automated flow line. In this

case, the simulation analyst tries to figure out the best buffer locations and the best buffer capacities according to these pre-selected performance measures.

Following the selection of performance measures, a decision is made about the assumptions under which the model will be developed. The assumptions involve to make a decision about the adequate system boundary. If this boundary is too wide, additional time and expense will be encountered in completing the simulation project. For example, let's assume that there is a manufacturing plant with two subdepartments, say, A and B (i.e., the parts are processed in department A and, then, in B). If the problem on hand is associated only with the department B, then there is no need to simulate the department A. If, on the other hand, the boundary is too narrow, the questions which are expected to be answered by a simulation study can not be accurately answered. Therefore, the assumptions decided on affect the complexity of the simulation model. The logic and the validity of the assumptions must be provable to reach accurate results.

After these two steps, the results obtained can be described via flow charts and the algorithm schemes. The flow charts and the algorithm schemes provide the visualization of the system logic and also ease the codification of the simulation model.

There are three alternative software tools to simulate a system. These are general purpose computer languages, special purpose simulation languages and dedicated simulators. Although, the general computer languages such as C, FORTRAN, PASCAL are flexible enough, they often require programming expertise to simulate complex interactions between the system components. If the complexity of the system to be simulated increases, the lead time to codify the simulation model increases. The simulation languages and the simulators are the most commonly used simulation tools. A simulation language is a computer package which is general in nature, but may have special features to model some manufacturing facilities such as workstations and material handling modules. GPSS, SIMAN, ARENA SE, and SLAMSYSTEM are some well known simulation languages. A manufacturing simulator is a computer

package which allows to simulate a system with little programming or in some cases with no programming (Banks&Gibson, 1997). Some of the simulators are PROMODEL, SIMFACTORY, WITNESS, etc. (Law&McComas, 1992, pp.29-34). The selection of a simulation tool is a very difficult task. The success of a simulation project is directly affected by this selection (Banks, 1991).

After the selection of simulation tool, the simulation model is codified based on the control logic described via flow charts and algorithm schemes. Verifying is the succeeding step in a simulation study.

2.2.4 Verifying The Simulation Model

After the development of the simulation model, the model needs to be verified. In simulation studies, verifying involves showing that the computer program performs as expected and intended, thus providing a correct logical and mathematical representation of the simulation model. Verifying is also called as debugging. In addition to simply eliminating the programming errors, the verification process also helps to see whether the code truly reflects the conceptual model. One of the goals of the verification is to show that all parts of the model work, both independently and together, and use the right data at the right time (Shannon et al., 1992, p.71). The techniques which are used to verify a simulation model are given as follows:

- Developing the program in a modular manner.
- Performing a structured walk-through of the code.
- Checking simulation output data for reasonableness.
- Using interactive debuggers and traces.
- Running the model for known characteristics.
- Animating the system .

Developing the simulation model in a modular manner facilitates the coding and debugging. It is impractical to debug the whole model. Subprograms and submodules

need to be controlled independently. If there is no problem in these submodules, this means that a new submodule or subprogram can be added to the main model. Developing the simulation model in a step by step manner reduces the chances for programming errors (Kleijnen, 1995).

When developing large simulation models, it is advisable to have more than one person checking the control logic of the computer program. Because the person who writes the program may get into a mental rut and thus, he/she may not be a good critic. In some organizations, this idea is implemented formally and is called a structured-walk through (Law et al., 1991, p.302)

Checking output data for reasonableness is a good way to figure out the logical or mathematical errors. The pilot run can be made after the model development. During the pilot run, the model is executed under a variety of input parameters and, then, the output is checked to see whether it is reasonable or not.

One of the most powerful techniques which can be used to debug a discrete event simulation program is to trace the control logic of the program. During the trace process, the state of the simulated system, the event list, the state variables, and the statistical counters, etc., are printed to a file just after each event occurrence. Following, this list can be checked to figure out errors and inappropriate cases. An interactive debugger allows an analyst to stop the simulation at a selected point in time, and to examine the values of certain variables (Law et al., 1991, p.303).

If a particular characteristic is known or can be calculated for a given input data, this can guide the simulation analyst to analyze the errors in the model. For example, deterministic numbers can be used to see whether the model behaves as we intend.

Sometimes, animation can be used for the purpose of verifying. However, using animation for verifying is not very confidential way. When the animation is used to verify a model, it needs to truly represent the intended system (Rohrer, 1996).

The next step in a simulation study is validation. Validation is one of the most important steps in simulation projects. This step helps to see whether the model developed accurately represents the real system.

2.2.5 Validating The Simulation Model

Validation should not need to be confused with the verification which was mentioned above. The verification does not guarantee that the model is valid. Following the verification of the simulation code, the control logic of the program is examined in detail to ensure that it truly represents the real system. Validation can be defined as to prove that the model which is completed can, indeed, be used for decision making. A validation procedure is concerned with three basic questions (Shannon et al., 1992, p.71):

- Does the model appropriately represent the real system?
- Are the output data which are generated by the simulation model similar to the operating characteristics of the real system?
- Does the user of the simulation model have confidence in the results of the model?

It must be noted that there are degrees of validation. Validation is not a decision variable indicating whether the model is valid or invalid. No one or two tests can validate a simulation model. Rather, confidence in the usefulness of a model must gradually increase as the model passes through more tests and as new points of correspondence between the model and the reality are found. Validation testing occurs continually in the process of designing, establishing, and using the model (Pegden et al., 1990, p.21).

One of the most important validation process is a three-step approach. This approach does not guarantee an absolutely valid model, but it will make the model more representative of the real system and also more credible. The first step is to develop the model with high face validity which means that the model seems

reasonable to the people who are knowledgeable about the system under study. In order to develop such a model, the simulation analysts should make use of all existing information including conversations with system experts, observations of the system, existing theories, adequate results from similar simulation models to compare and his experience (Law et al., 1991, pp.307-322).

- **Conversations with system experts:** Building a simulation model is not a type of task which can be carried out by one person. The simulation analyst needs to figure out all of the information sources requiring the necessary information for model building and to have conversation with system experts. Additionally, this information is very useful even if it is not used for simulation purposes.
- **Observations of the system:** The input data may be collected by using time study techniques. Extra care must be taken in this process to ensure that the data are correct, in the right format and representative of what is being modeled.
- **Existing theories:** The existing theories can be utilized for to support the validation of the model. If the theory says that the interarrival times are distributed exponentially, whatever the simulation analyst found is not important.
- **Relevant results from similar simulation models:** If the similar simulation models are available and the results of these simulation models are confidential, then the results of these models can be used for validating.
- **Experience:** Experience, perhaps, is one of the most important things which result in more valid models.

The second step in three-step approach is to test the assumptions of the model empirically. A lot of quantitative tests are available to ensure that the assumptions which were made are logical and representative of the real system. For example, if a theoretical distribution has been fitted to some observed data and used as input to the simulation model, the adequacy of the fit can be tested by the graphical plots and the goodness of fit tests such as Chi-Square and Kolmogorov-Smirnov. The homogeneity of the data which were observed can be tested by Kruskal-Wallis. For example, the interarrival times may be fitted to different theoretical distributions during a day. In

this case, the use of one theoretical distribution to represent interarrival times may not give confidential results. One of the most useful tools during the second step of validation is sensitivity analysis. This can be used for determining the input parameter which has the most important effect on the simulation output. This input parameter needs to be analyzed more carefully.

The third step in three-step approach is to determine how representative the simulation output data are. There is a number of statistical tests available to ensure that simulation output really represents the output of the actual system. However, this comparison is not simple as it appears. Since almost all the real systems and simulation output are non-stationary (the distributions of observations change over time) and autocorrelated, classical statistical tests can not be directly applicable. Some of the methods which can be used to compare the output of the model and the output of the system are as follows:

- **Inspection Approach:** Inspection approach involves comparison of one or more statistics from the real system and from the model without the use of formal statistical procedures. These statistics can be sample mean, sample variance, etc. In this approach, the number of the observations which are taken from both the real system and the model must be same. The disadvantage of this approach is that just one or two statistics obtained without formal statistical procedures can not be sufficient to conclude that the model is a good representative of the real system.
- **Correlated Inspection Approach:** This approach uses the same historical input data for both the real system and the model to obtain more reasonable comparison criteria.
- **Confidence-Interval Approach Based on Independent Data:** This approach tests whether the output data generated by the model significantly differ from the output data collected from the real system. In this approach, paired-t test and hypothesis testing are used to compare the real system and the model. This is a more reliable way to compare the model and the system output.

2.2.6 Design Of Simulation Experiments

After the model is validated, the experimentation conditions need to be prepared. In short, the design of experiments is a method for planning an experiment so that the necessary data can be collected and analyzed by statistical methods. The parameters which we want to adjust in an experiment are called factors which are set at different levels. The design of experiment prescribes a recipe for setting the factors and levels. When the simulation experiments are run and the response data are collected, the main effects of factors and the interaction of the factors can be determined. The simulation analyst, then, can essentially pick and choose the levels of the factors to achieve the desired response (Akhun&Durmuşoğlu, 1994). The design of experiments is a very important phase of a simulation study. While design of experiments provides the recipe for the experiment and a method for analyzing the results, simulation provides an environment to run the experiment. When it is impractical to experiment with the actual system, the design of experiments and the simulation can be combined to form a very powerful analysis tool (Porcaro, 1996, pp.24-30).

Experimental design supports the simulation analyst to decide which configurations to simulate so that the desired knowledge is extracted with minimum effort. In a model with only one factor, the design of experiments is simple: run 'n' replications at different levels of the factor, and determine if there is a significant difference between different levels of that factor. However, if the number of factors increases, the simulation analyst should have background in factorial design. In a factorial design, the actual effects of these factors on the system response can be studied. In these designs, all combinations of the factor levels are examined. This is called full factorial design. If the number of factors increases, the full factorial design requires a lot of experimentation and long computational time. Therefore, in these cases, the fractional factorial design can be used. This method requires the simulation analyst to know much about in statistical issues and design of experiments (Houshyar et al., 1993, p.56).

2.2.7 Statistical Analysis Of Outputs

The approach used to analyze the outputs of a simulation model depends on whether the system is terminating or non-terminating. A terminating system has a fixed starting condition to which the system returns after each termination and an event defining the natural end of the simulation. For example, the systems which work within the time interval of 8:00-17:00 are terminating systems. The fixed starting condition in this example is 'empty and idle' and the termination event is the close of the business day. A non-terminating system has neither a fixed starting condition nor a natural ending point. Examples of non-terminating systems are those that do not close such as a hospital or the manufacturing system which operates continuously 24 hours per a day (Pegden et al., 1990, pp.174-191). In general, most manufacturing systems tend to be non-terminating type because although they may close, they do not have a fixed starting condition to which they return. For example, the inprocess inventory from one day is typically carried over to the next day. The system is terminating if the arrivals are turned off, and the system is emptied before closing for the day. Since only the replication number is specified in a terminating system, analyzing the terminating system is easier than the analyzing the non-terminating system. Non-terminating systems are more sophisticated and complex to analyze and require good knowledge of statistics.

In a terminating system, both the starting conditions and the terminating conditions are defined by the nature of the system. Because the starting conditions and the replication length can not be manipulated, the only decision in controlling the sample size is the number of replications of the simulation to execute. Since the independence of the each replication can be provided by controlling the random number streams, it is possible to apply statistical procedures directly. This means that a particular observation in a particular replication is not affected by a particular observation within the previous replication. Therefore, the average of the statistics within the each replication can be used as the unique observations which are taken from the mass and they are normally distributed. In this case, the confidence interval on the performance measure can be constructed by using statistical methods. In

addition to this, it is possible to determine the additional number of replications required to reach given confidence level.

In a non-terminating system, there is no event that causes the system to return to a fixed initial condition. Accordingly, there is no natural basis for selecting either the starting conditions or the length of the run. In these systems, the important case is to understand the steady state behaviour of the system. The steady state behaviour of the system covers a very long period of time. Non-terminating systems generally go through an initial transient phase that varies with the starting conditions. Thereafter, they have an essentially unchanging distribution that is independent of the starting conditions. There is no definite point at which the system's behaviour changes from transient to steady state. Because we want to estimate the performance of the system during the steady state phase, any observations recorded during the transient phase will bias our results. This initial transient phase of the simulation run can be called as warm-up period. There are a lot of statistical techniques to figure out the point that the steady state behaviour of the system starts. Some of them are as follows: Gafarian's Method, Conway's Method, Fishman's first and second Method, Gordon's Method, and Tocher's Method. In addition to these, the plot diagrams can help to figure out this point in time. Another method to reduce the initial bias is to run the model for a long simulation period and to observe the time that the warm-up period ends. Following, the data which were observed during the warm-up period can be truncated to reduce the initial bias.

2.2.8 Variance Reduction Techniques

In simulation studies, the main idea behind the statistical analysis is to estimate the performance of the system. Hence, to use the right statistical analysis methods is very important. If the system to be simulated is a very complex one and the simulation period is very long, the simulation runs may take very long time. Furthermore, if the number of design points and the number of replications to be carried out at each design point is too large, the simulation may require days to execute. To avoid large number of replications and to obtain more reliable results, the methods which are

called as “Variance Reduction Techniques” (VRT) are used for obtaining efficient statistical results. The efficient statistical result, here, emphasizes decreasing the variance of the random output variable.

Although VRTs can save computer time by reducing the required number of replications of the model, in general, the use of these techniques makes the analysis complex. Their proper use requires skill and experience. Furthermore, there is generally no way to know in advance if the application of a VRT will actually reduce the variance. In some cases the use of VRTs can actually yield poorer results than a straightforward analysis. However, in most cases these procedures are effective at reducing the variance of the estimator. The most important VRTs are Indirect Measures, Control Variates, Common Random Numbers and Antithetic Variates. These methods are widely used in simulation analysis. However, Common Random Numbers technique is probably the most useful and the popular VRT of all.

In some cases, the performance measure to be analyzed can be calculated by analytical methods. There may be an analytical relationship involving the performance measure in question. This analytical formulas can be used to estimate the performance measure indirectly. Indirect estimate has a smaller variance than the direct estimate of this value. The use of the indirect estimates depends on the system being modeled and the prior knowledge about the system. For example, many simulation models measure performance as related to queuing. Queuing theory and its analytical relations are used for indirect estimate of the performance. In many cases, the indirect measure provides a tighter confidence interval, often leading to a 30 to 40 percent reduction in the variance. Since, in the case of indirect estimate, the fixed and known quantity is replaced by the random component, this reduction can be provided. For example, the total time spent at a station is the sum of two random components: the time spent waiting in the queue and the processing time. In the indirect estimate method, the second term is replaced by its known mean value, and it is, therefore, reasonable to expect a reduction in the variability of the overall estimate. The main disadvantage of Indirect Measures is that it may be difficult to find a known relation for a variety of situations. On the other hand, the main advantage of this method is its simplicity and

its ability to obtain both direct and indirect measures from the same run.

The second VRT is Control Variate. The method of Control Variate depends on the prior knowledge about the system. In this case, there is no need to know the exact mathematical relationship between the control random variable and the interested random variable. It is enough to know that the values are related. The control random variable represents the prior knowledge about the system such as the time between arrivals etc. For example, from replication to replication, the observed average time between arrivals, c_i , may be larger or smaller than its expected (known) value. When the observed interarrival time is larger than the expected value, it is expected that fewer parties arrive and the throughput will be slightly lower than expected. Likewise, when more parties than expected arrive, it is expected that the throughput will be higher than expected. Therefore, the random variable C (i.e., interarrival time) is said to partially control the random variable of throughput (X). The method can be formulated as follows:

$$x_{ci} = x_i - a(c_i - E(C)) \quad a^* = \text{Cov}(X, C) / s^2(C)$$

x_{ci} : Corrected observations of interested random variable

x_i : Observations of interested random variable

c_i : Observations of control random variable

$E(C)$: Expected value of control random variable (it is known)

a, a^* : weighting factor, optimal weighting factor

The corrected random variable X_c always has less variance than the directly observed random variable X , as long as an optimal weighting factor is used and correlation exists between the X and C . This method is very easy to use and like the method of Indirect Measures, the same set of replications can be used to calculate the variance with and without control variates.

Common Random Numbers is the most widely used VRT method since many of the simulation languages provide facilities to apply this method. This method is used when two or more alternative designs are compared with each other. The basic idea in

the method of Common Random Number is that any differences in performance are due to differences in the system configuration rather than due to fluctuations in the experimental conditions. In simulation, these experimental conditions are the generated random variates that are used to drive the models through simulated time. The term Common Random Numbers comes from the use of the same random numbers within the pairs of replications in order to induce the needed positive covariance between the responses. Simply using the same random number seed for each replication within a pair is usually not sufficient to induce covariance; It must be also ensured that random numbers are synchronized. Synchronization implies that the random numbers are used at the same junction and for exactly the same purpose across the system.

The last VRT is Antithetic Variates. Like Common Random Numbers, this technique reduces the variance by artificially inducing a correlation between replications of the model. Unlike Common Random Numbers, the Antithetic Variates Technique is applied when the single system performance needs to be estimated. In its simplest form, Antithetic Variates tries to induce negative correlation by using complementary random numbers to drive the two runs in a pair. That is, if U_k is a particular random number used for a particular purpose in the first run, $(1-U_k)$ needs to be used in the second run for the same purpose. Like Common Random Numbers, synchronization needs to be provided to induce correlation between the separate runs. Many of the simulation languages provide facilities to apply the Antithetic Variates techniques.

2.3 Software For Simulation

Two major types of simulation software for manufacturing applications are available. These are simulation languages and simulators. Simulation languages are described as having essentially unlimited modeling flexibility, but they require programming expertise. Simulators, on the other hand, are often much easier to use, but many simulators are not applicable to certain problems of interest. The general purpose computer languages can also be used to simulate a system. However, if the

system to be simulated is a very complex one, it is very difficult to simulate the system by using general purpose computer languages such as C, FORTRAN, PASCAL.

2.3.1 General Purpose Languages

General purpose computer programming languages can be used to simulate a system. This term “General Purpose” is self explanatory, that is, these languages can be used for solving any kind of problems. In other words, these are not designed for particular tasks. Some of the general purpose computer programming languages are BASIC, C, FORTRAN, and PASCAL, etc. Since these languages have the capability to generate random numbers, they can be used for simulation purpose.

In the case of using a general purpose computer language, all of the events which describe the behaviour of the system need to be codified. If the system to be simulated is a very complex one, to efficiently and accurately represent the relations between the system components can be quite troublesome through the general purpose languages. Since they are not special purpose languages, they do not have output analyzer, input analyzer, validation and verification capabilities. A successful simulation study requires simulation environment to have all these capabilities. Therefore, using a general purpose computer language may result in inflexible models and long development time, unless the simulation modeler is an expert in programming techniques. However, if the system to be simulated is not a complex one, general programming languages can be used as a simulation software. The advantages of the general computer programming languages can be summarized as follows (Law et al., 1991, p.236):

- Most modelers already know a general purpose language, but this is often not the case with simulation languages.
- FORTRAN, PASCAL, BASIC or C is available on virtually every computer. They are easily accessible.
- Efficiently written C or PASCAL programs may require less execution time.
- General purpose languages may allow greater programming flexibility.

- They are often more affordable than simulators and simulation languages.

2.3.2 Simulation Languages

There are currently two major classes of simulation software: simulation languages and simulators. A simulation language is a computer package that is general in nature but may have special features for certain types of applications. For example, SIMAN and SLAM II have manufacturing modules for conveyors and automated guided vehicles. A model is developed in a simulation language by writing a program using the modeling constructs of the language. Generally, these modeling constructs (i.e., blocks) include macro words such as queue, delay, release, generate, etc. These words are used for explaining the history of the entities as the entities pass through the system. When the entity enters a block, it executes the related routines which are, indeed, written in some general purpose computer language by the designers of that simulation language. The major advantage of the simulation languages is their ability to model almost any kind of system, regardless of the operating procedures or the control logic of the system to be modeled. Some of the other advantages of simulation languages are as follows:

- Simulation languages provide most of the capabilities such as validation, verification, output and input analysis to carry out a successful simulation study.
- They provide the natural language based macro words which make the simulation language easy to learn.
- Simulation models are generally easier to change when they are written in a simulation language.
- Most simulation languages provide dynamic storage allocation during execution. They provide automated error detection facilities since many potential types of errors have already been identified.

Some of the desirable features of simulation languages can be summarized as follows (Haider&Banks, 1986, pp.25-31):

- **Input Flexibility:** The simulation language design needs to have build-in flexibility, to minimize the input entering effort.
- **Syntax:** The syntax used in a simulation language needs to be user-friendly and needs to have the capability to represent system graphically to facilitate development of the model and to reduce errors made during model development.
- **Structural Modularity:** Simulation languages need to be designed in a modular manner. In such a modeling environment, each component of the model can be changed without altering other. Furthermore, each module can be compiled independently and linked only at the time of the model execution.
- **Modeling Flexibility:** As it is mentioned before, three types of modeling approaches are available. The process interaction approach has the most flexible structure. However, it is very difficult to represent every special case with the process interaction approach (Mazziotti, 1996).
- **Interface to Other Software:** If this interface is provided, the special cases can be carried out by using general purpose languages.
- **Standard Statistic Generation:** The simulation language needs to have the capability of statistics generation to facilitate the input and output analysis.
- **Animation:** Animation may be useful for rough analysis. For example, to see the bottleneck point in the system, animation can be used. However, animation is not an analysis tool, and it can not be substituted by statistical methods.
- **Interactive Model Debugging:** The simulation languages may have a variety of choices to execute the simulation model. For example, Step by step execution of the model provides better understanding and debugging conditions.
- **Cost:** The cost of the simulation software is one of the most important features since it directly affects the justification of the use of the tool.

The most important simulation languages are SIMAN, ARENA SE, AutoMod II, GPSS, SLAM II, SLAMSYSTEM, and SIMSCRIPT II.5. The selection of the simulation languages affects the success of a study. Therefore, the aim of the study needs to be specified carefully, and the requirements to achieve that aim must be

carried out with utmost care.

2.3.3 Simulators

A manufacturing simulator, in its most basic form, is a computer package that allows one to simulate a system contained in a specific class of manufacturing systems without programming. The particular system of interest is selected for simulation by choosing items from menus, by filling in forms or by the use of graphics. Typical modeling constructs for a simulator are parts, machines, queues, transporters and conveyors. The major advantage of a manufacturing simulator is that program development time may be considerably less than that for a simulation language. This may be very important given the tight time constraints in most manufacturing systems (Law et al., 1992, pp.29-30).

The simulators are easier to learn and they have modeling constructs which are more closely related to the system of interest. The major disadvantage of many simulators is that they are limited to modeling only those system configurations allowed by their standard features. This difficulty can be overcome, if the simulator contains programming-like commands to model complex decision logic or if the simulator has the ability to call routines written in a general purpose language. Selection of an appropriate simulator depends on the requirements of the problem, and the features of the simulators. In general, for a detailed analysis of an existing system, simulation languages are used, whereas for an aggregate analysis of a proposed system, simulators are preferred. Examples of simulators are, SIMFACTORY, STARCELL, WITNESS, XCELL+, AND PROMODEL.

CHAPTER THREE

SIMULATION FOR DECISION SUPPORT IN A VIDEO CASSETTE PRODUCTION PLANT

3. Simulation For Decision Support In A Video Cassette Production Plant

This thesis aims at simulating the assembly, the loading and the packaging operations of an electronics company which produces audio cassettes, video cassettes and compact discs for both domestic and international markets. Since each product group (audio cassettes, video cassettes and compact discs) has a distinct manufacturing process which is highly automated, it is possible to divide the system into a number of subsystems. Each resource such as machines, workers, material handling systems, etc., in these subsystems has been dedicated to a particular product group. Because of difficulties in simulating the whole system which consists of hundreds of operations, this thesis particularly focuses on the problems in video cassette manufacturing subsystem.

3.1 Problem Definition

One of the most important functions of the upper management in any manufacturing plant is to make sure that all of the actions to be done in the plant are in control. The control function of the upper management requires verifying whether the activities taking place in the plant are in conformity with the plan which is adopted, the instructions which are issued and the standards which are established. For example, the budget which is prepared by the budget department sets the standards for the money flow in the plant. Provided that the budget is prepared realistically (not too large, not too small), it can be used to control the money coming in or going out. If any department of the plant overspends its budget, the upper management tries to figure out how to avoid having this kind of situation in the

future. Therefore, the budgets of each department in the plant need to be prepared according to the realistic requirements of the departments.

Towards the end of every year, the budget planning department of the video cassette production plant starts to prepare the budget for the next year. At this stage, the budget department tries to decide on how and where the resources on hand (money) are transferred. The allocations for investment, income/expenses and the staffs are the three main considerations in budget preparation in the plant. Especially in staffs budget preparation, the budget department faces some problems. For example, the preparation of the budget for staffs requires the budget department to know about the staff requirements in each department. The staff requirements are determined by each department according to the predefined production goals.

Currently, the number of the staffs in each department is calculated by using deterministic methods. For example, the number of feeding operators is calculated based on the standard time. We assumed that an automated assembly line consists of 5 serial assembly machines. These assembly machines add the components “A, B, C, D, and E” to the base part (one unit is used for each assembly operation). The hoppers of the assembly machines are fed with these components in the party size of 100 units. The standard times required to feed the hoppers with 100 units are equal and this standard time is equal to 30 seconds. The speed of the line is one base part per second (the speed of the line is production target). Based on these data, the number of feeding operator is calculated as follows:

The time required to feed one unit of component = $30/100 \Rightarrow 0.3 \text{ sec./comp.}$

The operator is in charge of feeding 5 assembly machines $\Rightarrow 0.3 * 5 = 1.5 \text{ sec./comp.}$

The number of feeding operators = $1.5/1 \Rightarrow 1.5 \approx 2 \text{ operators}$ (“1” is the line speed)

The disadvantages of this method can be listed as follows:

- The stochastic behavior of the system is not taken into consideration.

- It ignores material handling aspects of the system.

This rough-cut approach for determining the staff needs is not supported by the upper management because they believe that the production departments are not realistic in specifying the staff requirements.

This thesis aims at evaluating the current case, and determining the staff requirements of each department for a given production target and setup condition by using the simulation as a decision support tool.

3.2 Overview Of The System

The video cassette manufacturing system consists of four main departments. These are plastic injection, assembly, loading and packaging departments. Figure 3.1 shows the layout of these departments. All of these departments operate 24 hours (3 shifts) in a day. The machines in these departments do not stop between the shifts.

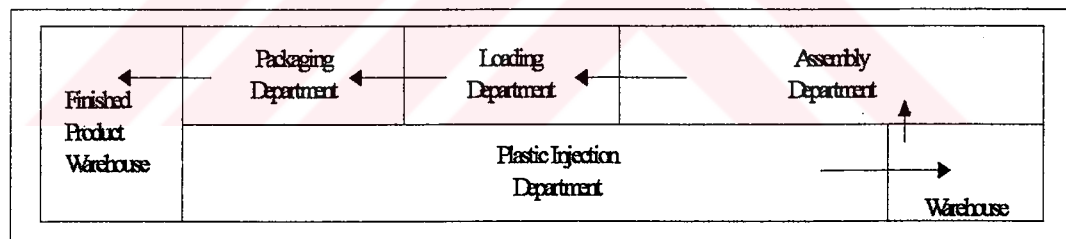


Figure 3.1 The General Layout Of The Departments

Two types of components are used for producing V0. These are plastic and steel components. Figure 3.2 represents the Bills of Material of a V0 (i.e., when all of the assembly operations are completed in the assembly department, the main structure of the video cassette without video tape is called V0). While the plastic components of the video cassette are produced in plastic injection department, the steel components are bought from other companies. All of the components which form a video cassette are stored in warehouse until a request is made.

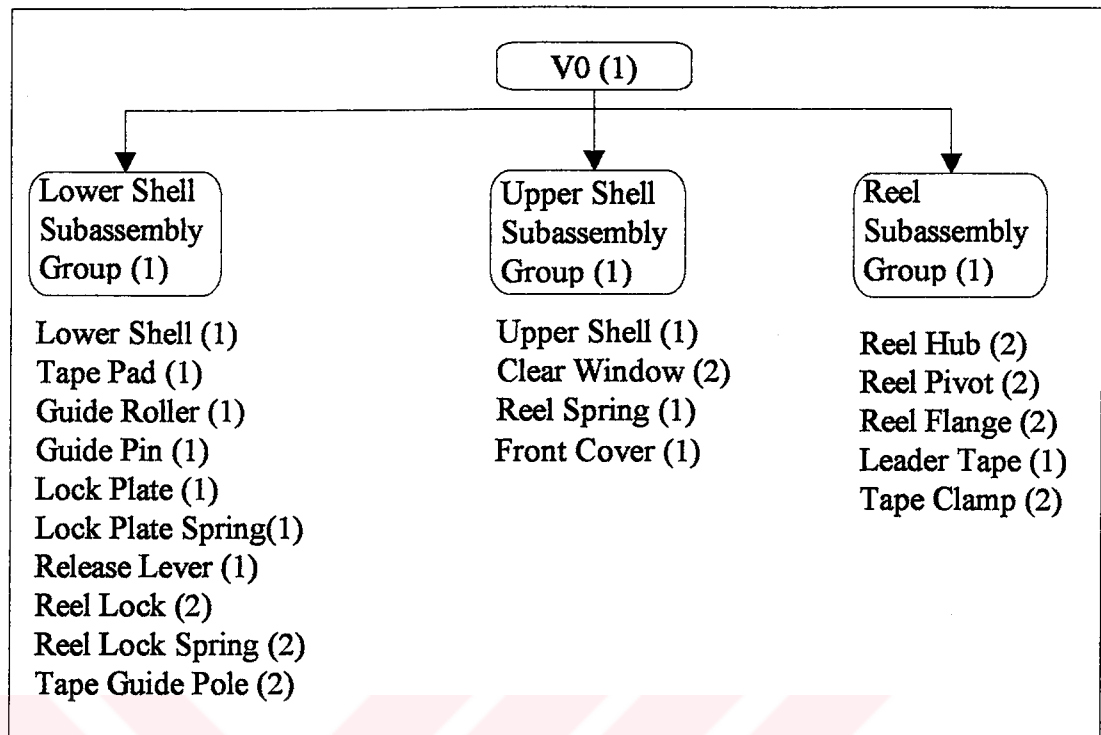


Figure 3.2 The Bills of Material of a V0

Due to the fact that the production rates of each plastic components are greater than the assembly lines speed and the steel components are available on time when they are requested, the warehouse works as an infinite buffer for the assembly department. This makes the assembly department independent from the plastic injection department, that is, there is no problem associated with the starving of the assembly department due to the nonexistence of the requested components in the warehouse. Therefore, the plastic injection department is excluded in this thesis.

After having completed all of the assembly operations, the V0s which are the main structure of the video cassette are sent in batches to the loading department. In loading department, V0s are loaded with video tape in specified length (according to the customer specification) by automated loading machines. The packaging and the loading department are connected to each other via conveyors. The loaded video cassettes are transferred to the packaging department by these conveyors. The packaging department is the last place where the finishing operations are done. The loaded video cassettes pass through all of the operations which are done by

automatically or manually. The finished products are then sent to the finished product warehouse which is next to the packaging department. Figure 3.3 shows the work flow between the departments. To transfer the work parts from one department to another, both hand trucks and a conveyor system are used. For example, the components are transferred from the warehouse to the assembly department by the hand trucks. However, the loaded VOs are transferred from the loading department to the packaging department by the conveyors. As it is depicted in Figure 3.3, there is a direct connection between each line in the loading department and the corresponding line in the packaging department.

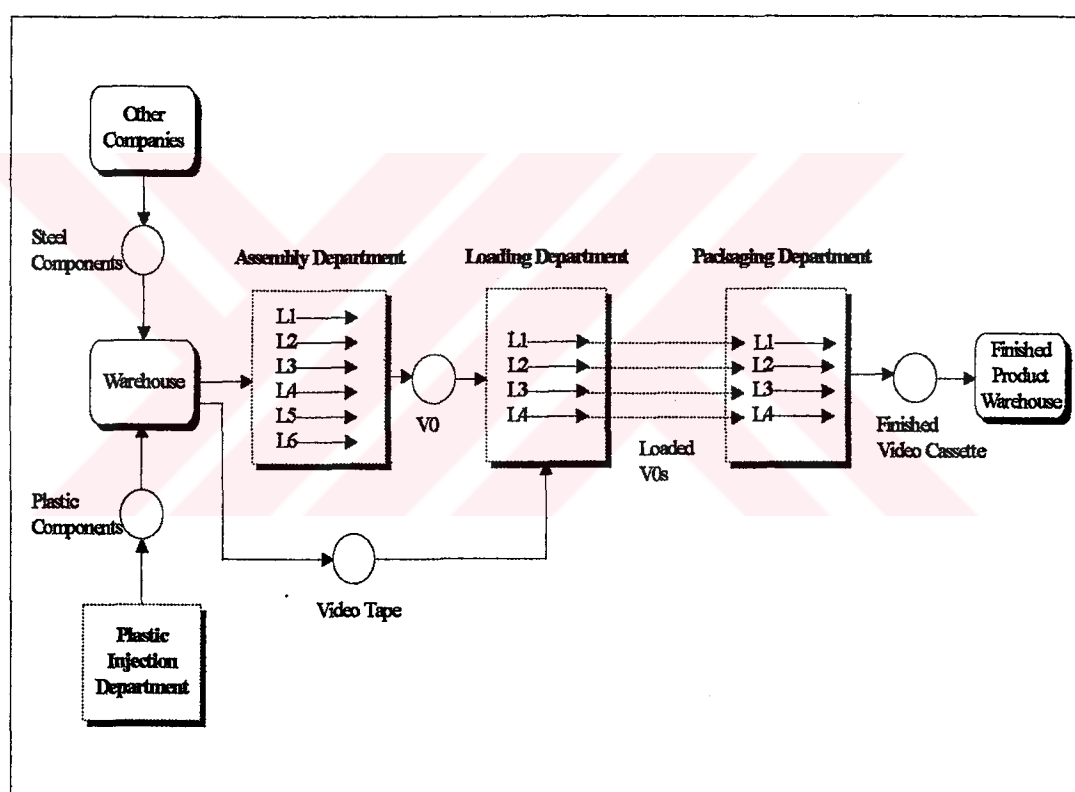


Figure 3.3 The General Work Flow Between The Departments

3.2.1 The Products

The video cassette manufacturing subsystem of the factory has the capability of producing four types of video cassettes. These video cassette types are VHS, BETA, 8 mm, and VHSC. This thesis deals with the video cassette type of VHS due to the

reason that the percentage of VHS cassettes produced in the last two years (1995-1996) was 96,62% of total production volume. Therefore, it can be assumed that all of the resources in the video cassette manufacturing subsystem are dedicated to the video cassette type of VHS. There are 742 kinds of products which were produced in the last two years as type VHS. These products differ from each other according to customer specifications such as video tape length, and the design features of package. Since all of the components to be assembled are same for all products, the assembly operations are not affected by the product differences. However, the loading and the packaging operations are affected by the product differences. For example, the loading operations require the video tape length as input. The video tape length may change from one product to another. These changes result in lower or higher loading time. Likewise, the package specifications may increase or decrease the time to pack a video cassette. Out of 742 kinds of products, 4 of them had the highest production volume in the last two years (e.g. 67.08% of the total production volume of VHS). Therefore, in this study, it is assumed that the video cassette manufacturing subsystem produces 4 kinds of products as type VHS.

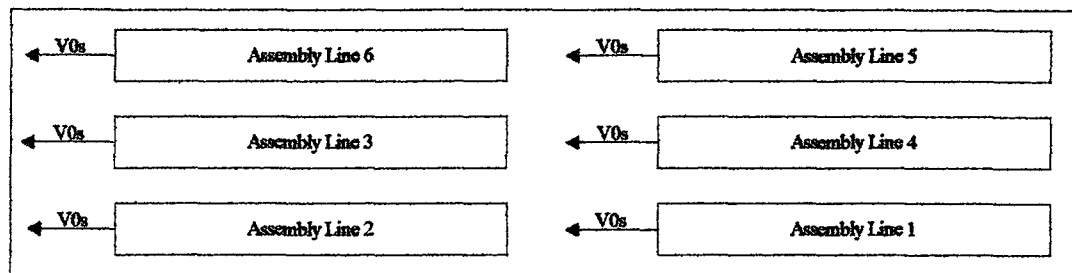
Table 3.1 shows the 4 types of products that were selected for the model. As it was mentioned before, the columns in this table show the specifications of the product type (i.e., video tape length, case capacity, and the large case capacity) that affect the production process. The measure unit for the Video Tape Length is in minutes. For example, "the video tape length 180" means that watching the whole video cassette takes 3 hours (the video tape length in 3-hour video cassette is 250 meters). The numbers in columns Case, and Large Case show the capacity of each package unit as the number of video cassettes. Since there are no differences between these four products in terms of the video tape length and the design feature of the package, these four products can be assumed that a single product with the video tape length of 180 minutes and the packaging type of "10 video cassettes in a case and 5 cases in a large case".

Table 3.1 The Four Products Having The Highest Production Volume

Product No.	Video Cassette Type	Product	Video Tape Length	Case	Large Case	Production Volume (1995-1996)
1	VHS	AQ	180	10	50	9,609,000
2	VHS	HQ	180	10	50	15,483,300
3	VHS	SQ	180	10	50	27,379,900
4	VHS	SG	180	10	50	12,248,900

3.2.2 The Assembly Department

The assembly department consists of six identical assembly lines. Except for the video tape, the steel and the plastic components are assembled to form a video cassette in each assembly line. Each line has 5 main assembly stations which consist of several automated assembly machines. Figure 3.4 shows the layout of the assembly lines in the assembly department. Each automated assembly machine has the hopper which loads the components into the automated assembly machines. Since the physical configuration of the automated assembly machines is dial-type in each main assembly station, these automated assembly machines are called dial indexing machines. In dial-type configuration, the base part to which the other components added is placed (loaded) in its nest which is attached to the indexing table. The components are assembled to the base part at the various assembly machines which are located around the indexing table.

**Figure 3.4 The Layout Of The Assembly Lines In The Assembly Department**

These assembly stations operate with intermittent motion, where the cycle time consists of the longest processing time plus the indexing time (Groover, 1992, pp.173-175). This type of motion makes the assembly machines in the same main assembly

station to depend on each other. Therefore, if an interruption caused by any reason (e.g. the nonexistence of the component to be assembled in the hopper, the failure of the automated assembly machine) occurs in any automated assembly machine, the indexing table of that assembly station stops. Figure 3.5 represents the main assembly stations, assembly machines and the conveyor system of an assembly line.

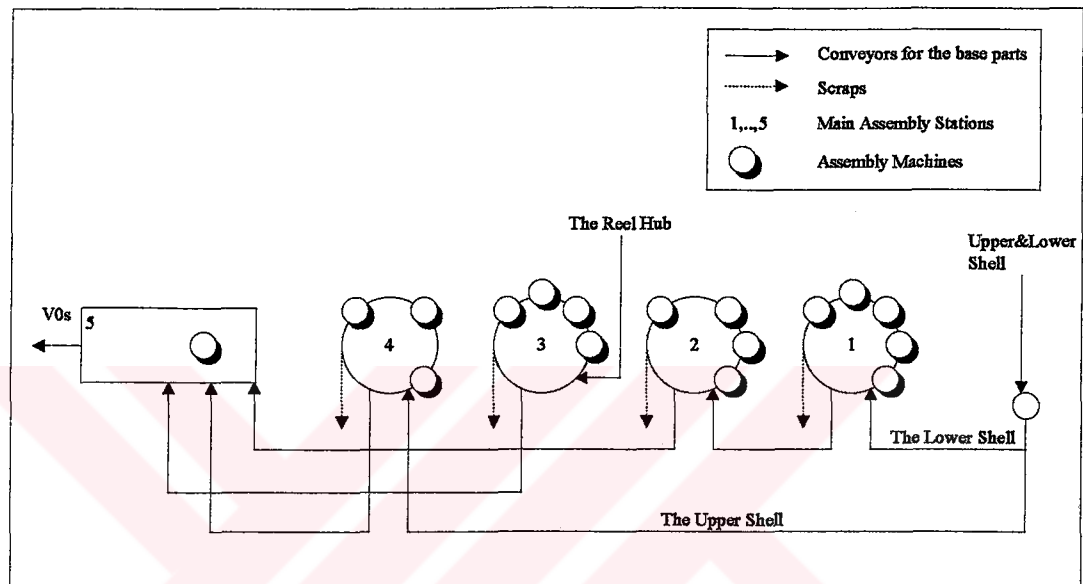


Figure 3.5 The Overview Of An Assembly Line

Each main assembly station has been designed for a particular base part in the assembly line. As a base part, the main assembly stations from 1 to 5 use lower shell, lower shell, reel hub, upper shell, and lower shell subassembly group, respectively. In these main assembly stations, the components are added to the base parts by the workheads of the automated assembly machines, then the base parts are sent to the next main assembly stations for the other assembly operations by the conveyors. All of the automated operations being done in an assembly line, their definitions and the work flow have been given in Table 3.2 and Figure 3.6, respectively.

Table 3.2 The List For The Operations Of An Assembly Line

Stat. No.	Main Assem. Station	Definitions
1	-	Upper&Lower Shell enters line from this station as a single unit
2	1	This station (robot) transfers the lower shell from conveyor to Main Assembly Station 1
3	1	Tape pad assembly (1-unit)
4	1	Guide roller assembly (1-unit)
5	1	Guide pin assembly (1-unit)
6	1	Lock plate assembly (1-unit)
7	1	Lock plate spring assembly (1-unit)
8	1	Inspection point for assembly operations in Main Assembly Station 1
9	1	The station where the failing lower shell subassembly is sent
10	1	Exit point of Main Assembly Station 1 for good lower shell subassembly
11	2	This station (robot) transfers the lower shell subassembly from conveyor to Main Assembly Station 2
12	2	Release lever assembly (1-unit)
13	2	Reel lock assembly (2-units)
14	2	Reel lock spring assembly (2-units)
15	2	Tape guide pole assembly (2-units)
16	2	Inspection point for assembly operations in Main Assembly Station 2
17	2	The station where the failing lower shell subassembly group is sent
18	2	Exit point of Main Assembly Station 2 for good lower shell subassembly
19	5	This station (robot) transfers the completed lower shell subassembly group from conveyor to Main Assembly Station 5
20	3	Reel hub enters line from this station as two units
21	3	This station (robot) transfers the reel hubs (2-units) from conveyor to Main Assembly Station 3
22	3	Pivot assembly (2-units)
23	3	Reel flange assembly (2-units)
24	3	Leader tape assembly (1-units)
25	3	Tape clamp assembly (2-units)
26	3	Inspection point for assembly operations in Main Assembly Station 3
27	3	The station where the failing reel subassembly is sent
28	3	Exit point of Main Assembly Station 3 for good reel subassembly
29	5	This station (robot) transfers the completed reel subassembly group from conveyor to Main Assembly Station 5
30	4	This station (robot) transfers the upper shell from conveyor to Main Assembly Station 4
31	4	Clear window assembly (2-units)
32	4	Reel spring assembly (1-units)
33	4	Front cover assembly (1-unit)
34	4	Inspection point for assembly operations in Main Assembly Station 4
35	4	The station where the failing upper shell subassembly group is sent
36	4	Exit point of Main Assembly Station 4 for good upper shell subassembly group
37	5	This station (robot) transfers the completed upper shell subassembly group from conveyor to Main Assembly Station 5
38	5	Assembly of completed lower shell, upper shell and reel subassembly groups
39	5	Exit point of Main Assembly Station 5 for good V0

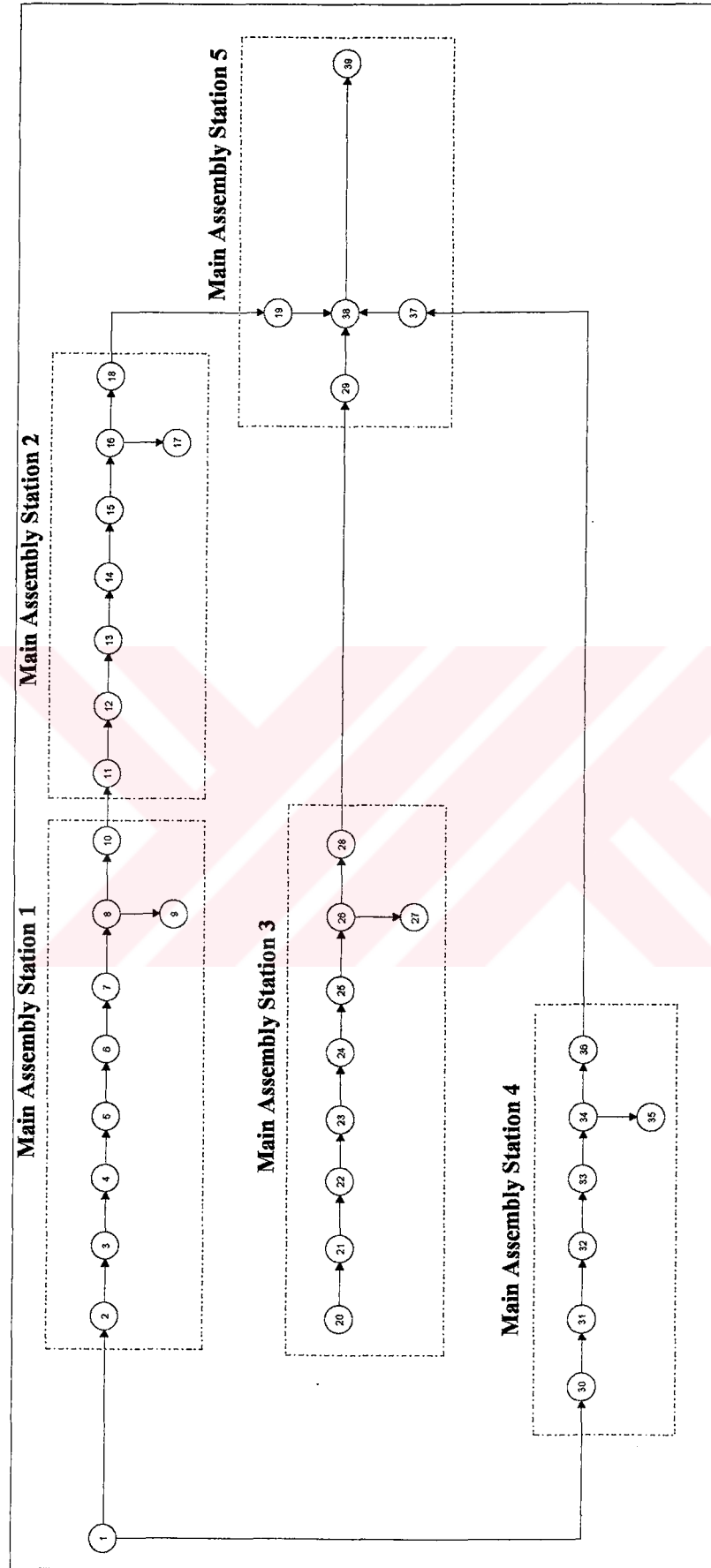


Figure 3.6 The Work Flow Between The Stations In An Assembly Line

Each main assembly station has a distinct inspection point. When the base part is transferred to the inspection point of a particular main assembly station by the indexing table, the components which were added to the base part are tested by the automated inspection machine, and with respect to the result of the inspection operation, the base part is sent to the next assembly station or the scrap station of that main assembly station. Since these are highly automated systems, a lot of data concerning with the operations are recorded automatically. For example, the number of the scraps or the number of the base parts which are successively passed from inspection operation can be seen on the monitor of the assembly station.

There are two types of operations in the assembly line. These are the operations which are done by the automated machines and the operations which are done manually by the workers. The processing times of the operations which are done by the automated machines are not stochastic. These are constant numbers and the processing time of each operation can be measured by a chronometer without statistical analysis (e.g. theoretical cycle time of each assembly station). However, the manual operations are stochastic. Specifying the realistic distribution for the processing time of an operation requires the use of statistics.

Two workers with hand trucks provide the connection between the warehouse and the assembly department. While the main assembly stations are operating, the number of components in each hopper decreases by the quantity assembled. When the number of the components is equal to a predefined warning level, a request is made for that component. If one of the transporters is idle, that transporter goes to the warehouse and brings the requested component to the storage area of that main assembly station. If both transporters are busy, the request is put in FIFO queue. All of the assembly lines are fed by these two transporters as explained above.

One technician and one worker are available for manual operations at each assembly line. The technician is assigned for the machine failures. If the assembly line stops because of some mechanical problems, the technician tries to find a solution.

The workers are assigned to the feeding operations. The components which are brought by the worker with hand truck in a specified quantity need to be fed by the worker to the hopper of each assembly machine.

3.2.3 The Loading Department

Following the assembly operations, the VOs are sent to the loading department in batches. The loading department consists of four parallel loading lines each having 20 loading machines. Figure 3.7 shows the layout of the loading department and the conveyor system. As in the case of assembly lines, these loading lines are highly automated. All of the loading machines operate independently. Any problem associated with any loading machine does not affect the whole line and the line keeps operating.

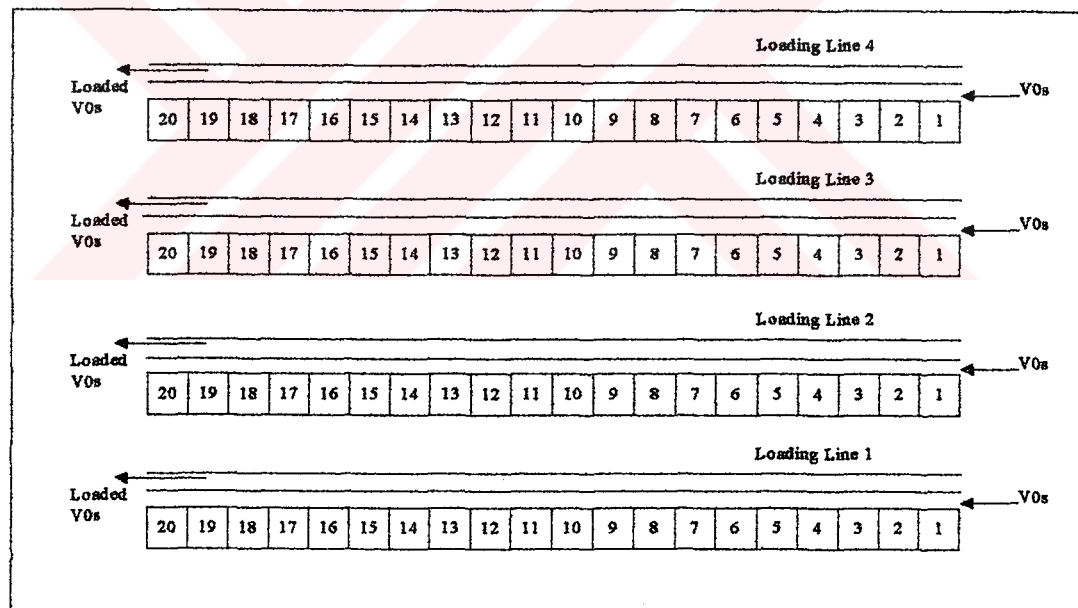


Figure 3.7 The Overview Of The Loading Department

As it is depicted in Figure 3.7, each line has two conveyors. The first conveyor is used for the transportation of the VOs. VOs are fed from the beginning of the loading line, then the conveyor transfers the VOs to the loading machine which has an empty space in its buffer. Each loading machine has the buffer with a capacity of 15 VOs. If

there is available space in the buffer of the loading machine, the sensor which is near to the entrance point of the loading machine allows the VOs to be entered. If the buffer of the loading machine is full, the entrance of the VOs is not allowed and VOs are transferred to the next loading machine. If the buffers of all loading machines in any loading line are full, the VO feeding operation and the first conveyor stops until the last loading machine is available. The second conveyor is used for transporting the loaded VOs. After the loading operation, the loaded VOs are ejected from the loading machine automatically and are sent to the second conveyor which is connected with the one of the packaging line of the packaging department.

The video tape is bought from the international markets as the pancakes which mean the reels on which the video tape has been rolled. These pancakes wait for the loading operations in the pancake buffer. These pancakes are attached to the loading machines by one pancake feeding operator for each line. Only one pancake can be attached to each machine.

Each loading machine has a warning system. The warning system of the loading machine consists of three lamps with the colours of green, yellow and red, respectively. The green light shows that there is no problem associated with the machine, and the machine keeps loading. The yellow light warns the pancake feeding operator that the video tape of that pancake is about to finish. The red light shows that the loading machine is not operating because of some malfunction. The possible reasons may be mechanical failures or empty pancakes. Since the loading machines are highly automated, the loading time for the same video tape length is constant. However, the video tape length differences between the products require different loading time.

3.2.4 The Packaging Department

The packaging department is the last place where the finishing operations are carried out. This department consists of four automated packaging lines and four manual packaging lines which are somehow continuation of each automated

packaging lines. As it is mentioned before, the automated packaging lines are connected with the loading lines by conveyors. Figure 3.8 shows the layout of the packaging lines.

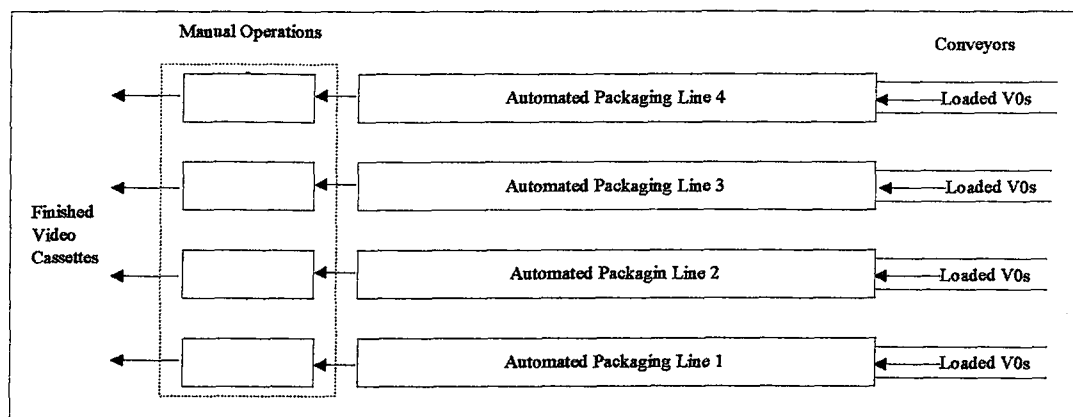


Figure 3.8 The Overview Of The Packaging Department

There are 8 serial operations which are carried out by the automated part of the packaging department. These operations which are carried out while the parts are on the conveyor prepare the video cassettes for casing and large casing operations. Table 3.3 shows these operations and their definitions.

Table 3.3 The Automated Operations In A Packaging Line

Station No.	Definitions
1	Labeling the loaded VOs
2	Coding the loaded VOs by ink jetting
3	Coding front cover of the VOs
4	Inserting
5	Putting the video cassette in a cartoon cover
6	Labeling the cartoon cover
7	Cellophaning
8	Shrinking

The manual operations which are carried out at the end of the automated part of the packaging lines limit the total video cassette production volume of the plant. Since the speed of the automated part of the packaging line is greater than the speed of the manual operations, there is a bottleneck point between the automated and the manual line. The manual operations are the casing and the large casing which consist of

several suboperations such as making a case or a large case, putting the video cassettes in a case or putting the cases in a large case, etc. These operations are affected by the customer requirements. For example, some of the customers do not require the case for the products which are produced in the video cassette production plant. If the case is not required, the casing operations are eliminated and the packs which consist of one or several video cassettes are put directly in a large case.

The number of the staffs who are assigned to the manual operations in the packaging department is not constant during a year because of the speed of the manual operations. Especially in the promotion periods, the operator requirements of the manual operations may change from day to day with respect to daily production load. In these periods, the direct labor needs are met from the indirect labors of the other departments.

3.3 Building The Model Of The Assembly Department

As it was mentioned before, the assembly department has 6 identical assembly lines each having 5 main assembly stations. The main assembly stations in an assembly line are connected with each other by a conveyor system. The main assembly stations from 1 to 5 consist of 5, 4, 4, 3, and 1 assembly machines, respectively.

The upper shells, the lower shells, and the reel hubs are the base parts and the line is fed with these base parts at a constant rate. The lower shells are transferred to the first main assembly station. Following 5 assembly operations on this station, they are transferred to the second main assembly station for the other 4 assembly operations, then they are sent to the fifth main assembly station for the last assembly operation. The upper shells are directly sent to the fourth main assembly station where 3 assembly operations are carried out, then they are transferred to the fifth main assembly station. The reel hubs are enter the line from the third main assembly station. After 4 assembly operations, to be combined with lower and upper shells, they are sent to the fifth main station.

Two kinds of conveyors are available in this department. The first one is the indexing tables (nonaccumulating) which transfers the base parts between assembly machines and the second one is the accumulating conveyor which transfers the base parts between the main assembly stations. There are 4 conveyors of type one and 8 conveyors of type 2.

Currently, each assembly line is operated by two workers. One of the workers is in charge of feeding the hoppers of the assembly machines. The second one is a technician and is in charge of handling the machine failures. Furthermore, to handle the transfer of components from the warehouse to the main assembly stations, two other workers are utilized. These two workers are in charge of the whole department and they serve as transporters. When a request is made by the assembly machines, these transporters bring the requested components in predefined batches from the warehouse. The feeding operator, then, feeds the hoppers with these components. The request for a component is made when the component level is equal to a predefined level which is called as warning level. This approach aims at decreasing the number of line stops caused by the lack of required components.

The objective of simulating this department is to evaluate the effects of changing the number of workers to be employed per line on daily throughput of a line, time-in-system (i.e., time-in-system means the total amount of time an entity spends in the line), the utilizations of workers, and the utilizations of main assembly stations. The pilot runs on simulation model of the assembly department having six lines showed that one replication took about 10 hours. Considering the fact that at least 5 to 10 replications are necessary to obtain reasonable results, an enormous amount of time was necessary to carry out experiments on the whole model. Therefore, to reduce the time requirements, the assembly department is evaluated as two subsystems each having three lines.

The factor which is the total number of workers to be employed per line is evaluated in three levels:

- 1 feeding operator, 1 technician, and 1 transporter per line.
- 1 feeding operator, 1 technician, and 1 transporter per 2 lines.
- 1 feeding operator, 1 technician, and 1 transporter per 3 lines.

To compare these alternative policies, three types of performance measures (i.e., utilization, throughput, and time-in-system) were used. These can be listed as follows:

- The daily throughput of the first assembly line.
- The utilization of the transporter.
- The utilization of the feeding operator.
- The utilization of the technician.
- The utilization of each main assembly station.
- Time-in-system.

3.3.1 Assumptions Of The Model

The assumptions made in building the model of the assembly department can be listed as follows:

- The upper shells, the lower shells, and the reel hubs are introduced into the system in batch size of 1.
- Feeding rates of the lower&upper shells, and the reel hubs are constant.
- These three base parts are always available at the beginning of the assembly line.
- The transporter is allowed to carry only one type of component per travel.
- The components are always available in the warehouse.
- The technician and the feeding operator remain at the last main assembly station that he visits.

3.3.2 Input Data Of The Model

Based on the assumptions given above, two groups of data are needed to be collected to build the model of the assembly department. The first group of data has deterministic nature and can be listed as follows:

- The batch size of lower&upper shells and the reel hubs.
- The time between arrivals of lower&upper shells and the reel hubs.
- The type and the speed of the conveyors.
- The speed of the feeding operator, technician and the transporter.
- The cycle times of each main assembly station.
- The batch size of each component.
- The number of components to be used in each assembly operation
- The warning levels of the hoppers of each assembly machine in a line.
- The distances between the warehouse and the main assembly stations (For transporter).
- The distances between the main assembly stations (For feeding operator and the technician).
- Lengths of the conveyor segments.

According to the list given above, the first deterministic data group is the batch sizes of the base parts. These base parts are introduced into the system in batch size of 1. At the beginning of each assembly line, there are main conveyors with a large capacity for the base parts. These conveyors, continuously feed the line with shells, and reel hubs at constant rates, that is there is no problem associated with the lack of the base parts. However, these main conveyors are excluded in the model because of the limited capacity of the computer on which the simulation model is run. Therefore, the time between arrivals of the base parts is assumed to be 2 seconds for each base part.

The third group of deterministic data is the speed and the type of each conveyor. Two kinds of conveyors are used in the system. These are accumulating and nonaccumulating conveyors. Indexing table is a kind of nonaccumulating conveyor. Except for the indexing tables, all of the conveyors are accumulating. In ARENA, conveyor, transporter speed and the length measure unit depend on the length of the entity. All these measures are determined according to the unit length of an entity. These values can not be real numbers. For example, in this thesis, the conveyor speed of 10 means that 10 video cassette length units per time. If the time unit is second and the video cassette length is 18 cm, the conveyor speed is equal to 1.8 m/sec in metric system. The speed of the indexing tables is equal to 10 video cassette length units per second. Although, the conveyor which transfers the finished products in the fifth main assembly station is not an indexing table (i.e., it is an accumulating conveyor), the speed of this conveyor is also 10 video cassette length units per second. The speed of the other accumulating conveyors is equal to 5 video cassette length units per second.

The speed of the worker, transporter, and the technician equal to 10 video cassette length units per second. This value does not change (constant) during the simulation run due to the reasons explained above.

Another group of deterministic data in assembly department is the cycle time of each main assembly station. The main assembly stations consist of assembly machines and the indexing table which transfers the base parts (lower shell, upper shell, or reel hub) according to the principles of synchronous transfer. The cycle time of each main assembly station includes transfer time to next station and the longest processing time of the assembly machines in that main assembly station. Table 3.4 represents the cycle time of each main assembly station in an assembly line. Since the assembly lines are identical, the cycle time of each main assembly station is valid for all of the assembly lines.

Table 3.4 The Cycle Times Of The Main Assembly Stations

Main Assembly Station	1	2	3	4	5
Cycle Time (sec./cass.)	2.15	2.15	2.00	1.95	1.88

The hopper of each assembly machine is fed with the related components in batches by the feeding operator. There are, totally, 16 assembly machines in four main assembly stations. There is no feeding operation in the fifth main assembly station. The batch size of each component can be seen in Table 3.5.

Table 3.5 The Batch Sizes Of The Components

Main Assembly Station No.	Assembly Machine No.				
	1	2	3	4	5
1	10.000	3.500	5.000	1.666	2.000
2	1.825	2.000	5.000	2.500	-
3	4.500	2.720	2.222	2.500	-
4	1.920	2.000	1.280	-	-

In Table 3.5, the batch sizes were given as the number of components in batches for the related main assembly station and the assembly machine number in that main assembly station. For example, the component which is assembled by the first machine of the first main assembly station is brought from the warehouse by the transporter in batch size of 10.000 units. Similarly, the hopper of this assembly machine is fed with this component by the feeding operator in batch size of 10.000 units.

After each assembly operation, the number of components in the hopper is decreased by the number of components to be assembled. Table 3.6 shows the number of the components to be used in each assembly operation.

Table 3.6 The No. Of Components To Be Used In Each Assembly Machine

Main Assembly Station No.	Assembly Machine No.				
	1	2	3	4	5
1	1	1	1	1	1
2	1	2	2	2	-
3	2	2	1	2	-
4	2	1	1	-	-

Based on this table, it can be said that the first machine of the first main assembly station assembles one unit of component in each cycle.

The warning level specifies the time that a request for component supply is needed to be placed so that the line operates smoothly. Since the transferring of the components from the warehouse requires time, the request which is made at the level of zero (i.e., no component in the hopper) causes the main assembly station to be stopped. Therefore, the warning level of each component should be determined at a realistic level. As a policy, it is assumed that the warning level of each component is equal to the 10% of the batch size of that component. Table 3.7 shows these levels.

Table 3.7 The Warning Levels Of The Components

Main Assembly Station No.	Assembly Machine No.				
	1	2	3	4	5
1	1.000	350	500	166	200
2	182	200	500	250	-
3	450	272	222	250	-
4	192	200	128	-	-

With respect to this table, it can be said that if the component level in the hopper of the first main assembly station of the first assembly machine decreases to the value of 1.000 components, the request is made for that component.

Since the transporter transfers the components between the warehouse and the main assembly stations, the distance set is defined as pairs. Table 3.8 shows the distances between the warehouse and the main assembly stations of each line. As it was mentioned before, the lengths have been given as a video cassette length unit.

As it can be seen in Table 3.8, the distance between the warehouse and the first main assembly station of the first line is equal to 80 video cassette length units. This length equals $18 \times 80 = 1440$ cm. $\Rightarrow$ 14.4 meters in metric system.

Table 3.8 The Dist. Bet. The Warehouse And The Main Assembly Stations

Assembly Line Number	Main Assembly Station No.			
	1	2	3	4
1	80	120	160	200
4	130	170	210	250
5	180	220	260	300

The distance set for the technician and the feeding operator is same and it is defined as the distance between each main assembly station of each assembly line. Table 3.9 shows the distance set for the technician and the feeding operator. This table represents the shortest distance of all alternative ways between each main assembly station.

Table 3.9 The Distances Between The Main Assembly Stations

Line No.	Main Assembly Station No.	1				2				3			
		1	2	3	4	1	2	3	4	1	2	3	4
1	1	-	40	80	120	50	90	130	170	100	140	180	220
	2	-	-	40	80	90	130	170	130	140	180	220	180
	3	-	-	-	40	130	170	130	90	180	220	180	140
	4	-	-	-	-	170	130	90	50	220	180	140	100
2	1	-	-	-	-	-	40	80	120	50	90	130	170
	2	-	-	-	-	-	-	40	80	90	130	170	130
	3	-	-	-	-	-	-	-	40	130	170	130	90
	4	-	-	-	-	-	-	-	-	170	130	90	50
3	1	-	-	-	-	-	-	-	-	-	40	80	120
	2	-	-	-	-	-	-	-	-	-	-	40	80
	3	-	-	-	-	-	-	-	-	-	-	-	40
	4	-	-	-	-	-	-	-	-	-	-	-	-

The last deterministic data group in the model is the length of the segments of the conveyors. ARENA enables modeling a conveyor by using the elements of segments. A segment implies a part of the whole conveyor between two following stations. For example, the conveyor which transfers the entities between 3 serial stations consists of two segments. The length of each segment is identified as a cassette length unit. If the length of a segment is 5 and at that time, five entities are being conveyed on this segment of the conveyor, the other entities waiting for transfer can not access that segment of the conveyor. As it was mentioned before, there are 12 conveyors in an

assembly line. Table 3.10 shows the lengths of the conveyor segments. “*” indicates that these conveyors are indexing tables.

Table 3.10 The Lengths Of The Conveyor Segments Of An Assembly Line

No.	From	To	Length
1	The enter point for the lower shells	The enter point of the first main assembly station	26
2	The exit point of the first main assembly station	The enter point of the second main assembly station	56
3	The exit point of the second main assembly station	The enter point of the fifth main assembly station	136
4	The enter point for the upper shells	The enter point of the fourth main assembly station	122
5	The exit point of the fourth main assembly station	The enter point of the fifth main assembly station	75
6	The enter point for the reel hubs	The enter point of the third main assembly station	20
7	The exit point of the third main assembly station	The enter point of the fifth main assembly station	75
8	The enter point of the fifth main assembly station	The exit point of the fifth main assembly station	3
9	The enter point of the first main assembly station	The exit point of the first main assembly station	7*
10	The enter point of the second main assembly station	The exit point of the second main assembly station	6*
11	The enter point of the third main assembly station	The exit point of the third main assembly station	6*
12	The enter point of the fourth main assembly station	The exit point of the fourth main assembly station	5*

In the assembly department, four stochastic data groups are necessary to build the model. The first one is the time between failures of any main assembly station in an assembly line. Except for the lack of component to be assembled, the term “failure” is used in this thesis as any reason which causes the main assembly station to stop. The engineers of the maintenance department have recorded these types of failures for a long time. However, these records can not be directly used as input in the model because of the principles of the management. Therefore, the input data used in the model relating to the time between failures and the probability that the main assembly station will be subject to failure are based on the estimations of the technicians. Table 3.11 shows stochastic behaviour of an assembly line. These values are valid for the other two assembly lines.

Table 3.11 The Time Between Failures In An Assembly Line

Main Assembly Station	The Probability of Where the Next Failure will Occur	The Time Between Failures (sec.)
1	0.25	Expo(1200)
2	0.25	
3	0.25	
4	0.25	

According to Table 3.11, it can be said that the time between failures in an assembly line is exponentially distributed with the mean of 1200 seconds and the next failure may occur in the first main assembly station with a probability of 0.25.

The second stochastic data group of an assembly line is the repairing time. As in the case of the time between failures, the repairing times used in the model are based on the estimations of the technicians. Since these kinds of estimations are poor to reflect the future and have large variability, the exponential distribution with the mean of 300 seconds was selected as a distribution.

The third stochastic data group is the probability of being scrapped. Each main assembly station (except for the fifth main assembly station) has its own inspection point. If the assembly operations are not conveniently made, the inspection machine in that main assembly station rejects these base parts, and removes from the system. The probability of being scrapped (i.e., 4.5%) is equal for all of the main assembly stations.

The last stochastic data group is the time to feed the hoppers of each assembly machine with components. When the transporter of the assembly department brings the order, the feeding operator feeds the related hopper of assembly machine. Each line has 4 main assembly stations (the fifth main assembly station assembles the subassemblies and does not require any components) and 16 assembly machines to be fed. 20 Observations for each hopper were taken per line and the best fit module of ARENA was used to find out which distribution represents best the observed data. The observations were taken with the 1/100 second digital chronometer. The tables from Table 3.12 to Table 3.27 represent the observation values of each hopper and the summary information about input data.

**Table 3.12 Curve Fit Summary For The Operation of *Feeding Tape Pad* In
The Assembly Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		14.63		11		15.22	
2		16.27		12		15.60	
3		16.56		13		18.37	
4		16.76		14		16.76	
5		17.99		15		17.04	
6		15.91		16		18.29	
7		12.59		17		17.89	
8		19.91		18		14.38	
9		15.70		19		20.33	
10		13.92		20		16.07	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	1	13.8	0.050	0.067	0.050	0.075
2	5	15.6	0.250	0.239	0.300	0.315
3	8	17.4	0.400	0.367	0.700	0.682
4	4	19.2	0.200	0.242	0.900	0.923
5	2	21.0	0.100	0.068	1.000	0.991

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	11s1rlfe.txt		Function	Square Error
Histogram Range	12 to 21		Normal	0.004247
No. of Data Points	20		Triangular	0.0046
No. of Intervals	5		Weibull	0.004831
Min. Data Value	12.6		Beta	0.01128
Max. Data Value	20.3		Gamma	0.01368
Sample Mean	16.5		Erlang	0.0149
Sample Std. Deviation	1.94		Lognormal	0.03046
Distribution Function	Normal		Uniform	0.075
Siman Usage	NORM(16.5,1.89)		Exponential	0.1534

**Table 3.13 Curve Fit Summary For The Operation of *Feeding Guide Roller*
In The Assembly Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		11.41		11		13.04	
2		13.15		12		12.44	
3		13.45		13		15.37	
4		13.67		14		13.67	
5		14.97		15		13.97	
6		12.76		16		13.46	
7		13.25		17		15.87	
8		12.06		18		11.14	
9		13.54		19		15.45	
10		10.66		20		12.94	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	2	11.30	0.100	0.060	0.100	0.069
2	3	12.48	0.150	0.200	0.250	0.269
3	8	13.65	0.400	0.328	0.650	0.597
4	3	14.83	0.150	0.268	0.800	0.866
5	4	16.00	0.200	0.109	1.000	0.975

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	11s1r2fe.txt		Function	Square Error
Histogram Range	10.1 to 16		Normal	0.03147
No. of Data Points	20		Weibulll	0.03203
No. of Intervals	5		Triangular	0.0326
Min. Data Value	10.7		Beta	0.03582
Max. Data Value	15.9		Gamma	0.03996
Sample Mean	13.3		Erlang	0.04019
Sample Std. Deviation	1.4		Uniform	0.055
Distribution Function	Normal		Lognormal	0.05703
Siman Usage	NORM(13.3,1.36)		Exponential	0.1303

**Table 3.14 Curve Fit Summary For The Operation of Feeding Guide Pin
In The Assembly Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		8.51		11		8.84	
2		9.43		12		9.05	
3		9.58		13		10.51	
4		9.70		14		9.70	
5		10.38		15		9.86	
6		9.22		16		11.21	
7		7.38		17		10.33	
8		10.01		18		8.37	
9		9.11		19		11.69	
10		8.12		20		9.31	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	1	8.00	0.050	0.060	0.050	0.066
2	4	9.00	0.200	0.238	0.250	0.304
3	9	10.00	0.450	0.381	0.700	0.685
4	4	11.00	0.200	0.245	0.900	0.930
5	2	12.00	0.100	0.063	1.000	0.993

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	l1s1r3fe.txt		Function	Square Error
Histogram Range	7 to 12		Normal	0.009723
No. of Data Points	20		Triangular	0.01255
No. of Intervals	5		Weibull	0.01418
Min. Data Value	7.38		Beta	0.02245
Max. Data Value	11.7		Erlang	0.02544
Sample Mean	9.52		Gamma	0.02845
Sample Std. Deviation	1.03		Lognormal	0.05039
Distribution Function	Normal		Uniform	0.095
Siman Usage	NORM(9.52,1.01)		Exponential	0.18

**Table 3.15 Curve Fit Summary For The Operation of *Feeding Lock Plate*
In The Assembly Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		17.10		11		17.74	
2		18.88		12		18.16	
3		19.19		13		21.16	
4		19.42		14		19.41	
5		20.75		15		19.72	
6		18.49		16		21.33	
7		14.89		17		20.64	
8		22.92		18		16.83	
9		18.26		19		20.29	
10		16.33		20		18.67	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	1	15.86	0.050	0.042	0.050	0.045
2	3	17.65	0.150	0.187	0.200	0.232
3	9	19.43	0.450	0.357	0.650	0.590
4	5	21.22	0.250	0.292	0.900	0.882
5	2	23.00	0.100	0.102	1.000	0.984

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	l1s1r4fe.txt		Function	Square Error
Histogram Range	14.1 to 23		Normal	0.01178
No. of Data Points	20		Weibull	0.01413
No. of Intervals	5		Triangular	0.0176
Min. Data Value	14.9		Erlang	0.02688
Max. Data Value	22.9		Gamma	0.0269
Sample Mean	19		Beta	0.02827
Sample Std. Deviation	1.91		Lognormal	0.04705
Distribution Function	Normal		Uniform	0.1
Siman Usage	NORM(19,1.86)		Exponential	0.1824

Table 3.16 Curve Fit Summary For The Operation of *Feeding Lock Plate Spring* In The Assembly Department

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		12.70		11		13.21	
2		14.12		12		13.54	
3		14.36		13		15.92	
4		14.54		14		14.54	
5		15.59		15		14.78	
6		13.80		16		16.43	
7		10.95		17		15.51	
8		15.10		18		12.49	
9		13.62		19		17.61	
10		12.09		20		13.94	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	1	11.82	0.050	0.052	0.050	0.057
2	4	13.37	0.200	0.227	0.250	0.284
3	9	14.91	0.450	0.386	0.700	0.669
4	5	16.46	0.250	0.257	0.950	0.926
5	1	18.00	0.050	0.067	1.000	0.993

SUMMARY REPORT			BEST FIT SUMMARY		
Data File		l1s1r5fe.txt	Function		Square Error
Histogram Range		10.3 to 18	Normal		0.005203
No. of Data Points		20	Weibull		0.009577
No. of Intervals		5	Triangular		0.0114
Min. Data Value		11	Beta		0.01628
Max. Data Value		17.6	Erlang		0.02598
Sample Mean		14.2	Gamma		0.02632
Sample Std. Deviation		1.57	Lognormal		0.04767
Distribution Function		Normal	Uniform		0.11
Siman Usage		NORM(14.2,1.53)	Exponential		0.1887

**Table 3.17 Curve Fit Summary For The Operation of Feeding Release Lever
In The Assembly Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	20.41	11	21.37
2	21.15	12	23.20
3	18.00	13	19.45
4	15.38	14	20.89
5	23.47	15	23.19
6	20.24	16	22.21
7	22.56	17	17.48
8	21.05	18	20.34
9	25.35	19	19.51
10	21.44	20	21.30

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	1	17.2	0.050	0.043	0.050	0.047
2	2	19.4	0.100	0.202	0.150	0.248
3	11	21.6	0.550	0.375	0.700	0.624
4	5	23.8	0.250	0.281	0.950	0.905
5	1	26.00	0.050	0.084	1.000	0.989

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	lls2r1fe.txt	Function	Square Error
Histogram Range	15 to 26	Normal	0.04297
No. of Data Points	20	Weibull	0.05664
No. of Intervals	5	Triangular	0.06469
Min. Data Value	15.4	Beta	0.07191
Max. Data Value	25.4	Erlang	0.09974
Sample Mean	20.9	Gamma	0.1062
Sample Std. Deviation	2.27	Lognormal	0.1582
Distribution Function	Normal	Uniform	0.18
Siman Usage	NORM(20.9,2.21)	Exponential	0.2656

**Table 3.18 Curve Fit Summary For The Operation of Feeding Reel Lock
In The Assembly Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		14.58		11		14.12	
2		18.07		12		15.46	
3		16.33		13		16.97	
4		16.51		14		18.51	
5		17.63		15		19.77	
6		15.74		16		20.61	
7		16.74		17		17.53	
8		17.27		18		14.35	
9		15.55		19		17.74	
10		17.94		20		17.89	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	3	15.4	0.150	0.136	0.150	0.171
2	6	16.8	0.300	0.288	0.450	0.459
3	8	18.2	0.400	0.311	0.850	0.770
4	1	19.6	0.050	0.172	0.900	0.942
5	2	21.00	0.100	0.049	1.000	0.991

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	11s2r2fe.txt		Function	Square Error
Histogram Range	14 to 21		Normal	0.02582
No. of Data Points	20		Triangular	0.0344
No. of Intervals	5		Weibull	0.03773
Min. Data Value	14.1		Beta	0.04686
Max. Data Value	20.6		Erlang	0.05081
Sample Mean	17		Gamma	0.05631
Sample Std. Deviation	1.7		Uniform	0.085
Distribution Function	Normal		Lognormal	0.09506
Siman Usage	NORM(17,1.66)		Exponential	0.1234

**Table 3.19 Curve Fit Summary For The Operation of Feeding Reel Lock
Spring In The Assembly Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	42.55	11	47.42
2	52.76	12	41.63
3	44.66	13	53.39
4	55.30	14	54.30
5	49.20	15	46.20
6	38.59	16	59.63
7	49.10	17	48.88
8	58.45	18	37.75
9	41.94	19	56.61
10	54.30	20	48.12

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	2	41.60	0.100	0.091	0.100	0.120
2	5	46.20	0.250	0.207	0.350	0.327
3	5	50.80	0.250	0.283	0.600	0.610
4	5	55.40	0.250	0.233	0.850	0.843
5	3	60.00	0.150	0.116	1.000	0.958

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	11s2r3fe.txt	Function	Square Error
Histogram Range	37 to 60	Normal	0.004468
No. of Data Points	20	Weibull	0.008131
No. of Intervals	5	Beta	0.01235
Min. Data Value	37.8	Uniform	0.02
Max. Data Value	59.6	Gamma	0.02017
Sample Mean	49	Erlang	0.02377
Sample Std. Deviation	6.49	Triangular	0.02766
Distribution Function	Normal	Lognormal	0.04266
Siman Usage	NORM(49,6.33)	Exponential	0.08767

Table 3.20 Curve Fit Summary For The Operation of Feeding Tape Guide Pole In The Assembly Department

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		20.46		11		25.06	
2		22.14		12		21.46	
3		22.44		13		24.30	
4		22.65		14		22.65	
5		23.91		15		22.94	
6		25.77		16		27.29	
7		18.36		17		23.81	
8		26.91		18		20.20	
9		24.99		19		26.31	
10		19.73		20		21.94	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	2	20.00	0.100	0.077	0.100	0.093
2	4	22.00	0.200	0.220	0.300	0.313
3	7	24.00	0.350	0.323	0.650	0.636
4	4	26.00	0.200	0.246	0.850	0.882
5	3	28.00	0.150	0.096	1.000	0.978

SUMMARY REPORT			BEST FIT SUMMARY		
Data File		11s2r4fe.txt	Function		Square Error
Histogram Range		18 to 28	Normal		0.006577
No. of Data Points		20	Triangular		0.0086
No. of Intervals		5	Beta		0.009191
Min. Data Value		18.4	Weibull		0.009808
Max. Data Value		27.3	Erlang		0.0219
Sample Mean		23.2	Gamma		0.02221
Sample Std. Deviation		2.45	Uniform		0.035
Distribution Function		Normal	Lognormal		0.0436
Siman Usage		NORM(23.2,2.39)	Exponential		0.1066

Table 3.21 Curve Fit Summary For The Operation of *Feeding Pivot* In The Assembly Department

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		12.61		11		10.21	
2		11.29		12		10.60	
3		13.58		13		13.43	
4		9.79		14		15.79	
5		13.05		15		12.08	
6		12.91		16		14.42	
7		14.53		17		12.94	
8		12.43		18		9.35	
9		10.70		19		15.44	
10		12.88		20		11.08	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	3	10.40	0.150	0.098	0.150	0.123
2	4	11.80	0.200	0.232	0.350	0.356
3	7	13.20	0.350	0.306	0.700	0.662
4	4	14.60	0.200	0.224	0.900	0.886
5	2	16.00	0.100	0.091	1.000	0.977

SUMMARY REPORT			BEST FIT SUMMARY		
Data File		11s3r1fe.txt	Function		Square Error
Histogram Range		9 to 16	Normal		0.006346
No. of Data Points		20	Triangular		0.008499
No. of Intervals		5	Beta		0.009271
Min. Data Value		9.35	Weibull		0.01268
Max. Data Value		15.8	Erlang		0.02292
Sample Mean		12.5	Gamma		0.02425
Sample Std. Deviation		1.82	Uniform		0.035
Distribution Function		Normal	Lognormal		0.04323
Siman Usage		NORM(12.5,1.78)	Exponential		0.08618

**Table 3.22 Curve Fit Summary For The Operation of *Feeding Reel Flange*
In The Assembly Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		28.56		11		30.53	
2		34.08		12		43.82	
3		35.03		13		41.12	
4		35.72		14		35.72	
5		39.86		15		36.67	
6		32.84		16		50.93	
7		21.70		17		39.52	
8		49.67		18		27.70	
9		32.15		19		47.72	
10		26.16		20		33.40	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	2	27.00	0.100	0.091	0.100	0.113
2	5	33.00	0.250	0.225	0.350	0.338
3	6	39.00	0.300	0.307	0.650	0.645
4	4	45.00	0.200	0.232	0.850	0.877
5	3	51.00	0.150	0.097	1.000	0.973

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	11s3r2fe.txt		Function	Square Error
Histogram Range	21 to 51		Normal	0.004602
No. of Data Points	20		Weibull	0.005158
No. of Intervals	5		Beta	0.008307
Min. Data Value	21.7		Triangular	0.0106
Max. Data Value	50.9		Erlang	0.01061
Sample Mean	36.1		Gamma	0.0137
Sample Std. Deviation	7.79		Uniform	0.025
Distribution Function	Normal		Lognormal	0.03464
Siman Usage	NORM(36.1,7.59)		Exponential	0.09249

**Table 3.23 Curve Fit Summary For The Operation of Feeding Leader Tape
In The Assembly Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	53.77	11	45.05
2	41.14	12	46.54
3	42.25	13	49.28
4	43.04	14	38.04
5	47.82	15	44.14
6	29.72	16	60.61
7	49.85	17	47.43
8	59.16	18	34.78
9	38.92	19	50.90
10	34.00	20	40.36

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	3	35.40	0.150	0.091	0.150	0.112
2	4	41.80	0.200	0.236	0.350	0.348
3	7	48.20	0.350	0.320	0.700	0.667
4	4	54.60	0.200	0.228	0.900	0.896
5	2	61.00	0.100	0.086	1.000	0.981

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	11s3r3fe.txt	Function	Square Error
Histogram Range	29 to 61	Normal	0.006629
No. of Data Points	20	Triangular	0.008486
No. of Intervals	5	Beta	0.0124
Min. Data Value	29.7	Weibull	0.01256
Max. Data Value	60.6	Erlang	0.02267
Sample Mean	44.8	Gamma	0.02457
Sample Std. Deviation	7.97	Uniform	0.035
Distribution Function	Normal	Lognormal	0.04635
Siman Usage	NORM(44.8,7.77)	Exponential	0.08586

**Table 3.24 Curve Fit Summary For The Operation of *Feeding Tape Clamp*
In The Assembly Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		19.55		11		20.81	
2		23.08		12		21.64	
3		23.69		13		27.58	
4		24.13		14		24.13	
5		26.77		15		24.74	
6		22.29		16		33.85	
7		15.17		17		26.55	
8		33.05		18		19.00	
9		18.10		19		31.80	
10		18.02		20		22.64	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	3	18.80	0.150	0.117	0.150	0.150
2	5	22.60	0.250	0.248	0.400	0.399
3	6	26.40	0.300	0.299	0.700	0.697
4	3	30.20	0.150	0.203	0.850	0.900
5	3	34.00	0.150	0.078	1.000	0.978

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	11s3r4fe.txt		Function	Square Error
Histogram Range	15 to 34		Normal	0.009116
No. of Data Points	20		Beta	0.01103
No. of Intervals	5		Weibull	0.01147
Min. Data Value	15.2		Triangular	0.01557
Max. Data Value	33.8		Uniform	0.02
Sample Mean	23.8		Erlang	0.0204
Sample Std. Deviation	5.03		Gamma	0.02091
Distribution Function	Normal		Lognormal	0.05096
Siman Usage	NORM(23.8,4.91)		Exponential	0.07409

**Table 3.25 Curve Fit Summary For The Operation of Feeding Clear Window
In The Assembly Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	80.72	11	79.11
2	94.97	12	80.97
3	97.09	13	110.64
4	98.61	14	94.61
5	107.75	15	100.73
6	92.23	16	132.35
7	67.52	17	107.03
8	129.51	18	92.83
9	90.69	19	125.24
10	77.41	20	93.47

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	3	80.20	0.150	0.116	0.150	0.151
2	5	93.40	0.250	0.250	0.400	0.400
3	6	106.60	0.300	0.301	0.700	0.701
4	3	119.80	0.150	0.203	0.850	0.905
5	3	133.00	0.150	0.077	1.000	0.982

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	11s4r1fe.txt	Function	Square Error
Histogram Range	67 to 133	Normal	0.009371
No. of Data Points	20	Beta	0.0104
No. of Intervals	5	Weibull	0.0115
Min. Data Value	67.5	Triangular	0.01557
Max. Data Value	132	Uniform	0.02
Sample Mean	97.7	Erlang	0.0204
Sample Std. Deviation	17.3	Gamma	0.02123
Distribution Function	Normal	Lognormal	0.05316
Siman Usage	NORM(97.7,16.9)	Exponential	0.07408

**Table 3.26 Curve Fit Summary For The Operation of Feeding Reel Spring
In The Assembly Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	19.57	11	21.99
2	24.53	12	22.91
3	25.21	13	29.58
4	25.71	14	25.32
5	28.68	15	26.39
6	23.64	16	36.62
7	15.65	17	28.43
8	35.72	18	19.96
9	17.14	19	34.32
10	18.85	20	25.34

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	3	19.40	0.150	0.115	0.150	0.149
2	5	23.80	0.250	0.247	0.400	0.396
3	6	28.20	0.300	0.300	0.700	0.696
4	3	32.60	0.150	0.205	0.850	0.901
5	3	37.00	0.150	0.079	1.000	0.980

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	11s4r2fe.txt	Function	Square Error
Histogram Range	15 to 37	Normal	0.009322
No. of Data Points	20	Weibull	0.009579
No. of Intervals	5	Beta	0.01058
Min. Data Value	15.6	Triangular	0.01556
Max. Data Value	36.6	Gamma	0.01606
Sample Mean	25.3	Erlang	0.01993
Sample Std. Deviation	5.81	Uniform	0.02
Distribution Function	Normal	Lognormal	0.03235
Siman Usage	NORM(25.3,5.66)	Exponential	0.07343

**Table 3.27 Curve Fit Summary For The Operation of *Feeding Front Cover*
In The Assembly Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	74.07	11	94.45
2	83.51	12	74.66
3	85.15	13	93.57
4	86.33	14	86.32
5	93.40	15	87.95
6	81.40	16	88.35
7	62.34	17	92.82
8	110.25	18	72.61
9	80.21	19	106.86
10	69.97	20	82.35

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	2	71.80	0.100	0.097	0.100	0.117
2	5	81.60	0.250	0.254	0.350	0.372
3	7	91.40	0.350	0.331	0.700	0.703
4	4	101.20	0.200	0.215	0.900	0.918
5	2	111.00	0.100	0.070	1.000	0.988

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	11s4r3fe.txt	Function	Square Error
Histogram Range	62 to 111	Normal	0.00153
No. of Data Points	20	Triangular	0.001967
No. of Intervals	5	Beta	0.009265
Min. Data Value	62.3	Weibull	0.01222
Max. Data Value	110	Gamma	0.03323
Sample Mean	85.3	Erlang	0.03644
Sample Std. Deviation	11.7	Uniform	0.045
Distribution Function	Normal	Lognormal	0.08236
Siman Usage	NORM(85.3,11.4)	Exponential	0.1123

To fit a distribution to the observed data “The Input Analyzer -Best Fit Module of The ARENA” was used. ARENA supports 9 distributions, Normal, Triangular, Beta, Weibull, Gamma, Uniform, Erlang, Lognormal, and Exponential distributions. For most of the distributions supported in the Input Analyzer, the curve fitting is based on the use of maximum likelihood estimators. Exceptions to this rule are Beta, Triangular and Uniform distributions. The Beta distribution is fit in two different ways: 1. using maximum likelihood estimators, 2. the method of moments. The results corresponding to the best of these fits are then retained. Triangular and Uniform distributions use the empirical rules to fit the distribution to the observed data.

As it is seen in the tables from Table 3.12 to Table 3.27, the comparison criterion for the best fit is square error. The square error is obtained from the data in columns “Probability Density-Data and Function”. The square error is equal to the sum of the squares of differences between columns “Data” and “Function”. The column “Data” shows the relative frequency of the data in related interval. Furthermore, the column “Function” shows the area under the fitted distribution between the lower and upper bound of the related interval. With respect to the tables from Table 3.12 to Table 3.27, the summary results of this analysis can be seen in Table 3.28.

Table 3.28 The Input Models Used For Feeding Operations

Stat.	Assembly		Machine	No.	
No.	1	2	3	4	5
1	norm(16.5,1.89)	norm(13.3,1.36)	norm(9.52,1.01)	norm(19,1.86)	norm(14.2,1.53)
2	norm(20.9,2.21)	norm(17,1.66)	norm(49,6.33)	norm(23.2,2.39)	-
3	norm(12.5,1.78)	norm(36.1,7.59)	norm(44.8,7.77)	norm(23.8,4.91)	-
4	norm(97.7,16.9)	norm(25.3,5.66)	norm(85.3,11.4)	-	-

According to this table, for example, the time to feed the first assembly machine of the first main assembly station is normally distributed with the mean and the standard deviation of 16.5 and 1.89, respectively.

3.3.3 Definition Of The Attributes, Variables, And Elements

In this section, the attributes, the variables, and the elements of the model will be defined. The term “L1” at the beginning of each definition implies the loading line number. This term takes the form of “L2”, and “L3” for the second, and third assembly lines, respectively. After the term “L1”, the term “s1” implies the main assembly station number. For example, L1s1, L1s2, L1s3 and L1s4 represent the information about the first, second, third and the fourth main assembly station of the first line. The attributes, variables, and elements used in this model can be listed as follows:

- **L1i:** It is a control attribute and stores the assembly machine number in the first main assembly station for the first assembly line. It takes values from 1 to 5.
- **L1n:** It is a control attribute and stores the assembly machine number in the second main assembly station for the first assembly line. It takes values from 1 to 4.
- **L1k:** It is a control attribute and stores the assembly machine number in the third main assembly station for the first assembly line. It takes values from 1 to 4.
- **L1l:** It is a control attribute and stores the assembly machine number in the fourth main assembly station for the first assembly line. It takes values from 1 to 3.
- **L1signal1(L1i):** It is a control attribute and controls the entity flow for the L1ith assembly machine of the first main assembly station.
- **L1signal2(L1n):** It is a control attribute and controls the entity flow for the L1nth assembly machine of the second main assembly station.
- **L1signal3(L1k):** It is a control attribute and controls the entity flow for L1kth assembly machine of the third main assembly station.
- **L1signal4(L1l):** It is a control attribute and controls the entity flow for the L1lth assembly machine of the fourth main assembly station.
- **Timein:** It is a control attribute and stores the time when the base parts are

introduced into the first assembly line.

- **abc:** It is a control attribute and used as an index for the set “stock”. It takes values from 1 to 4. It represents main assembly station number. If other two lines are added to the model it takes values from 1 to 12 (3 lines*4 main assembly station per line).
- **Stock(abc):** It represents the abc_{th} member of the set “Stock”.
- **L1s1s, .., L1s4s / L4s1s, .., L4s4s / L5s1s, .., L5s4s:** The members of the set “Stock”.
- **L1s1inv(L1i):** It is a variable and stores the component level of $L1i_{th}$ assembly machine in the first main assembly station. It takes the form of “L1s2inv(L1n)”, “L1s3inv(L1k)”, and “L1s4inv(L1l)” for the second, third and the fourth main assembly stations in the first assembly line.
- **L1s1bat(L1i):** It is a variable and stores the batch size of the component which is assembled at the $L1i_{th}$ assembly machine in the first main assembly station. It takes the form of “L1s2bat(L1n)”, “L1s3bat(L1k)”, and “L1s4bat(L1l)” for the second, third and the fourth main assembly stations in the first assembly line.
- **L1s1warning(L1i):** It is a variable and stores the warning level of $L1i_{th}$ assembly machine in the first main assembly station. It takes the form of “L1s2warning(L1n)”, “L1s3warning(L1k)”, and “L1s4warning(L1l)” for the second, third and the fourth main assembly stations in the first assembly line.
- **L1s1u(L1i):** It is a variable and stores the number of the component to be assembled in each assembly operation for $L1i_{th}$ assembly machine in the first main assembly station. It takes the form of “L1s2u(L1n)”, “L1s3u(L1k)”, and “L1s4u(L1l)” for the second, third and the fourth main assembly stations in the first assembly line.
- **L1s1batchsizea:** It is a variable and represents the arrival batch size of the lower shells. Its initial value is 1. If the number of the lower shells to be accessed to the conveyor exceeds 10, the value of “0” is assigned to “L1s1batchsizea” until this number decreases to a predefined level.
- **L1s1batchsizeb:** It is a variable and represents the arrival batch size of the upper shells. Its initial value is 1. If the number of the upper shells to be

accessed to the conveyor exceeds 10, the value of “0” is assigned to “L1s1batchsizeb” until this number decreases to a predefined level.

- **L1s3batchsize:** It is a variable and represents the arrival batch size of the reel hubs. Its initial value is 1. If the number of the reel hubs to be accessed to the conveyor exceeds 10, the value of “0” is assigned to “L1s3batchsize” until this number decreases to a predefined level.
- **L1s1processtime:** The cycle time of the first main assembly station. It takes the form of “L1s2processtime”, “L1s3processtime”, and “L1s4processtime” for the second, third and the fourth main assembly station, respectively.
- **L1enters1:** The name of the station (In ARENA, the transfer of the entities is provided by the term “station”. Therefore, each assembly machine has a unique station name) where the lower&upper shells are introduced into the system.
- **L1enters2:** The name of the station where the reel hubs are introduced into the system.
- **L1end:** The name of the station where the finished VOs are transferred.
- **Warehouse:** The name of the station where the components are brought by the transporter.
- **L1s1load:** The loading station of the first main assembly station. The name of this station takes the form of “L1s2load”, “L1s3load”, and “L1s4load” for the second, third and the fourth main assembly station, respectively.
- **L1s1end:** The name of the exit station for the first main assembly station. The name of this station takes the form of “L1s2end”, “L1s3end”, and “L1s4end” for the second, third and the fourth main assembly station, respectively.
- **L1s1bad:** The name of the inspection station for the first main assembly station. The name of this station takes the form of “L1s2bad”, “L1s3bad”, and “L1s4bad” for the second, third and the fourth main assembly station, respectively.
- **L1s7:** The final assembly operation is made in this station (fifth main assembly station).
- **L1s6load1, .. , L1s6load3:** The lower shells, the upper shells, and the reel hubs enter the fifth main assembly station from these stations.

- **L1s1(L1i):** The name of the station set which shows the $L1i_{th}$ member of that set. It takes the form of “L1s2(L1n)”, “L1s3(L1k)”, and “L1s4(L1l)” for the second, third and the fourth main assembly stations in the first assembly line.
- **L1s1r1, .. , L1s1r5:** The members of set “L1s1”.
- **L1s2r1, .. , L1s2r4:** The members of set “L1s2”.
- **L1s3r1, .. , L1s3r4:** The members of set “L1s3”.
- **L1s4r1, .. , L1s4r3:** The members of set “L1s4”.
- **L1s1r(L1i):** The name of the assembly machine set which shows the $L1i_{th}$ member of that set. It takes the form of “L1s2r(L1n)”, “L1s3r(L1k)”, and “L1s4r(L1l)” for the second, third and the fourth main assembly stations in the first assembly line.
- **L1s1r1_r, .. , L1s1r5_r:** The members of set “L1s1”.
- **L1s2r1_r, .. , L1s2r4_r:** The members of set “L1s2”.
- **L1s3r1_r, .. , L1s3r4_r:** The members of set “L1s3”.
- **L1s4r1_r, .. , L1s4r3_r:** The members of set “L1s4”.
- **L1s1end_r, .. , L1s4end_r:** The robots which transfer the base parts from main assembly stations to the conveyors.
- **L1s1load_r, .. , L1s4load_r:** The robots which transfer the base parts from conveyors to the main assembly stations.
- **L1s1bad_r, .. , L1s4bad_r:** The name of the inspection machines.
- **L1s1c, .. , L1s4c:** The name of the indexing tables.
- **L1c2:** The conveyor which transfers the lower shells from the station “L1enters1” to the station “L1s1load”.
- **L1c3:** The conveyor which transfers the upper shells from the station “L1enters1” to the station “L1s4load”.
- **L1c6:** The conveyor which transfers the lower shells from the station “L1s1end” to the station “L1s2load”.
- **L1c8:** The conveyor which transfers the reel hubs from the station “L1s3end” to the station “L1s6load3”.
- **L1c10:** The conveyor which transfers the lower shells from the station “L1s2end” to the station “L1s6load1”.

- **L1c11:** The conveyor which transfers the lower shells from the station “L1s4end” to the station “L1s6load2”.
- **L1c12:** The conveyor which transfers the reel hubs from the station “L1enters2” to the station “L1s3load”.
- **L1c13:** The conveyor which transfers the base parts from the stations “L1s6load1”, “L1s6load2”, “L1s6load3” to the station “L1s7”.
- **L1s1c_seg, .. , L1s4c_seg:** The segment set of the indexing tables.
- **L1c2_seg:** The segment set of the conveyor “L1c2”.
- **L1c3_seg:** The segment set of the conveyor “L1c3”.
- **L1c6_seg:** The segment set of the conveyor “L1c6”.
- **L1c8_seg:** The segment set of the conveyor “L1c8”.
- **L1c10_seg:** The segment set of the conveyor “L1c10”.
- **L1c11_seg:** The segment set of the conveyor “L1c11”.
- **L1c12_seg:** The segment set of the conveyor “L1c12”.
- **L1c13_seg:** The segment set of the conveyor “L1c13”.
- **tr1:** The name of the transporter.
- **tr1_dst:** The distance set of the “tr1”.
- **technician:** The name of the technician.
- **worker:** The name of the feeding operator.
- **tech_dst:** The distance set of the “worker”, and “technician”.
- **tr1_q:** The name of the queue of “tr1”.
- **workerq:** The name of the queue of “worker”.
- **technicianq:** The name of the queue of “technician”.
- **L1c2_q:** The name of the queue of “L1c2”.
- **L1c3_q:** The name of the queue of “L1c3”.
- **L1c12_q:** The name of the queue of “L1c12”.
- **L1s6load1_q, .. , L1s6load3_q:** The names of the queues of “L1s6load1_r”, “L1s6load2_r”, and “L1s6load3_r”.
- **L1end_c:** It is a counter and stores the total number of VOs produced.

Based on the explanations made so far, the general entity flow in a loading line can

be summarized in Figure 3.9. The SIMAN model and experiment listing files for an assembly line can be seen in Appendix A.1 and Appendix A.2, respectively.

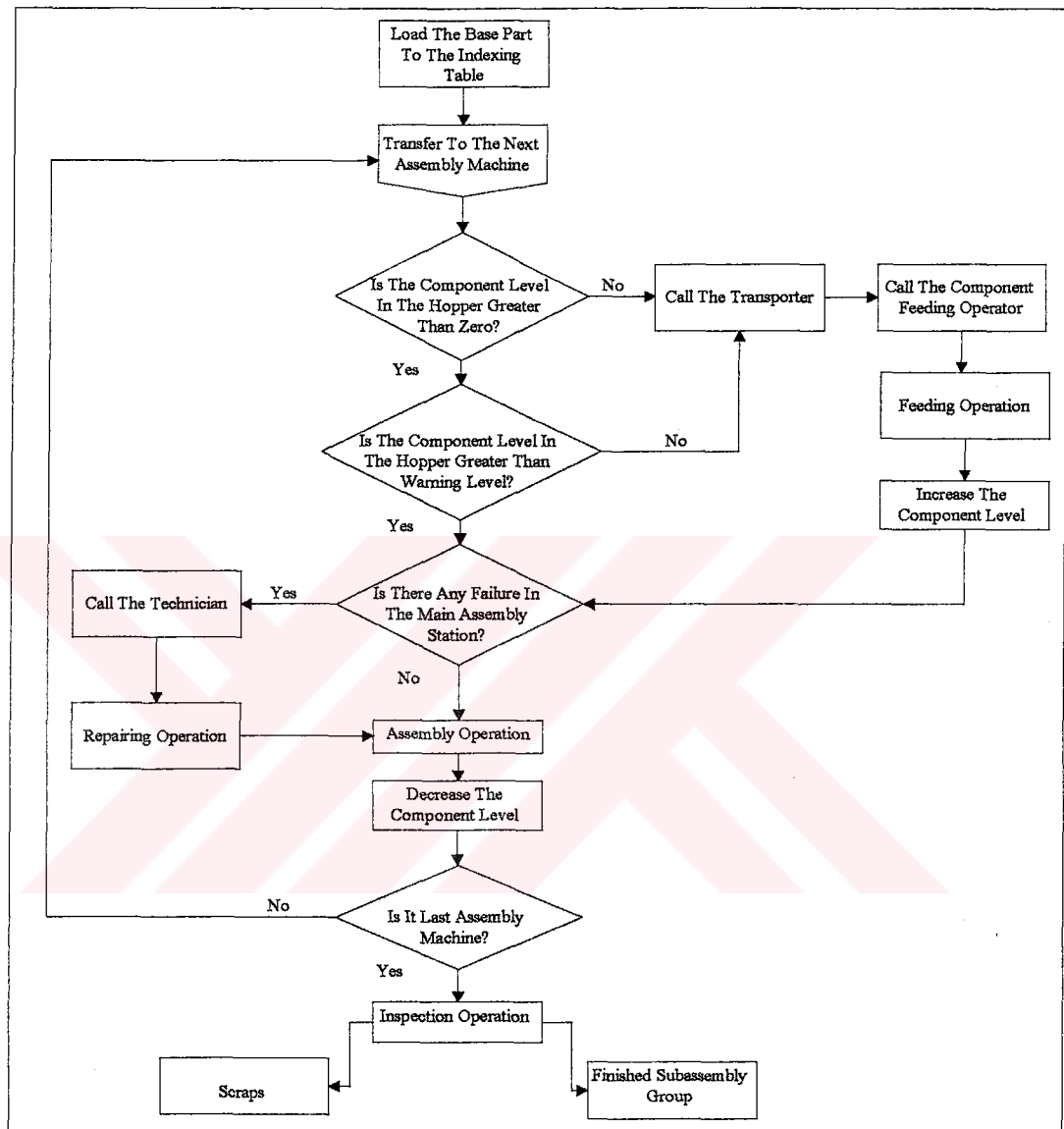


Figure 3.9 The General Algorithm For A Main Assembly Station

3.3.4 Verifying The Model

Following the model building, several verification methods have been used to make sure that the model works as we intended. ARENA provides some facilities that ease the verification of the model. First of all, it provides a modular program development

environment. Modular programming enables step by step verification while the model is being built. This is a very useful feature because it may be very difficult to verify the whole model.

The second method that has been used in model verifying is animation. For example, a part of the model can be executed for the special conditions and the animation can be observed to compare the logic of entity flow with expectations. Animation is a good verification technique but it should not be used alone. It should be supported by other verification techniques.

The third technique that has been used to verify this model is to execute the model for a special condition for which we can calculate the values of the response variables analytically. For example, if single entity is created and all of the stochastic variables are assumed to be deterministic, it is possible to calculate the total amount of time that a single entity requires until all of the specified operations are completed. The results from the analytical calculation and the results from the simulation can be compared to verify a model.

3.3.5 Validating The Model Of An Assembly Line

Since the assembly department operates continuously for 24 hours, the methodology of analyzing nonterminating systems is used. In analyzing nonterminating systems, the first step is to make a pilot run long enough and to collect one or two types of statistics for deciding on the truncation point. In the validation process, the first loading line is evaluated for daily throughput. As it was mentioned before, three methods can be applied to reduce the initial bias on the performance measures that we are interested in:

- Discard the data which are collected during the transient phase.
- Select an appropriate initial condition.
- Run the simulation long enough that any data collected during the transient phase is dominated by data collected during the steady-state phase.

The approach used to determine the truncation point in this study is the first one, the statistics to be collected for generating a time plot graph is “time-in-system”, and the selected replication length of the pilot run is 86400 seconds (24 hours). During the pilot run, the flowtime (time-in-system) of each V0 is saved in a file.

To analyze these observed data, “The Output Analyzer of ARENA” has been used. Firstly, Moving Average Module was used to identify the initial conditions. The purpose of using moving average module is to see the starting point of the steady-state behaviour of the system. Truncation point of the system is selected as $t=1000$ seconds (total observations=31307). This means that if we truncate these data, we decrease the initial bias caused by transient phase. Since the performance measure of the validation process is the daily throughput of the first line, 5 independent replications (the system and the statistics are initialized at the end of each replication) with the length of the $86400+1000=87400$ seconds were made. 1000 seconds can be called as warm-up period. The 5 independent daily throughputs of the first assembly line can be seen in Table 3.29. Since the hardware capacity is limited, the number of replications is reduced to five.

Table 3.29 The Simulation Results For The First Assembly Line

Replication Number	V0s (Daily Throughput)
1	31433
2	28860
3	30860
4	30098
5	29550

Based on current setup conditions of the assembly lines, the long period-analysis shows that the capacity of the first assembly line is 30000 V0s per day. This expectation can be compared with the results obtained from the simulation output to prove that the model is valid. Since these are independent observations, t-test can be applied. The hypotheses to be tested has been expressed as follows:

$$H_0 : \mu=30000 \text{ V0s per the first line}$$

$$H_1 : \mu \neq 30000$$

Critical t-value for $\alpha=0.05$ and the degree of freedom D.F=4 $\Rightarrow t_{n-1, 1-\alpha/2}=2.776$

$\bar{x} = 30160$ VOs per the first line

$S_{\bar{x}} = 456.97$

t-statistics $\Rightarrow (\bar{x} - \mu) / S_{\bar{x}} = 0.3506 \Rightarrow 0.3506 < 2.776$ H_0 is accepted.

Since the H_0 hypothesis is accepted, it can be said that the mean of the simulation output for daily throughput is not different from the expected value in %95 confidence level.

3.3.6 The Results Of Experimental Studies For The Assembly Department

After building and validating the model, the experiment plan was designed. As it was mentioned before, only three lines are evaluated because of the limited hardware capacity. The factor “the number of total workers employed per line” is evaluated in three levels. Table 3.30 summarizes the factor and its levels to be evaluated for the assembly department.

Table 3.30 The Factor Plan For The Assembly Department

Factor	Factor Levels		
	1	2	3
The number of workers employed per assembly line	1 feeding operator, 1 technician and 1 transporter per assembly line	1 feeding operator, 1 technician and 1 transporter per 2 assembly lines	1 feeding operator, 1 technician and 1 transporter per 3 assembly lines

In the current case, two transporters provide the connection between 6 assembly lines and the warehouse (In this study, only three lines are evaluated). During the plant trips, it has been observed that transporters are not heavily utilized. One transporter is enough to keep three lines operating. This means that the differences between the performance values of the alternative policies are not originated from the transporter but the feeding operator and the technician. To evaluate these alternative policies, it should be decided on the truncation point of each policy. Therefore, two

additional pilot runs have been made to understand steady state behaviour of the alternative policies. According to these pilot runs, the replication conditions summarized in Table 3.31 have been obtained.

Table 3.31 The Replication Conditions For The Assembly Department

Conditions	Factor Levels		
	1	2	3
Warm-up Period	1000	2000	2000
Replication Length	87400	88400	88400
No. of Replications	5	5	5
Initialize System	yes	yes	yes
Initialize Statistics	yes	yes	yes

Based on the experimental conditions given above, to evaluate each policy, 5 independent replications were made and the simulation results (i.e., daily throughput of the first line) summarized in Table 3.32 were obtained.

Table 3.32 Throughputs Of The First Assembly Line For Each Factor Level

Rep.No	Factor Levels		
	1	2	3
	Throughput	Throughput	Throughput
1	31433	27583	26672
2	28860	27648	23591
3	30860	30436	25769
4	30098	29697	24046
5	29550	29120	24554

Based on Table 3.32, it is clear that the third policy is the worst one in comparison to the other two policies. However, to decide on whether the difference between the first and the second policy is statistically important, a statistical analysis needs to be made. The statistical approach to be used to compare these policies is t-test. The hypothesis to be tested is that there is no difference in daily throughput between these two policies.

$$H_0 : \mu_D = 0 \text{ (No difference in daily throughput)}$$

$$H_1 : \mu_D \neq 0$$

Critical t-value for $\alpha=0.05$ and the degree of freedom D.F=4 $\Rightarrow t_{n-1, 1-\alpha/2}=2.776$

$\bar{D} = 1263.4$ VOs per the first line (the average of differences)

$S_{\bar{D}} = 664.68$

t-statistics $\Rightarrow (\bar{D} - \mu_D) / S_{\bar{D}} = 1.901 \Rightarrow 1.901 < 2.776$ H_0 is accepted.

Since the H_0 hypothesis is accepted, it can be said that the daily throughput of the first policy is not different from the daily throughput of the second policy in %95 confidence level. However, the critical t-value affects the acceptance of the H_0 hypothesis. This value tends to decrease while the degree of freedom is increasing. Therefore, it must be noted that H_0 hypothesis may not be accepted if the number of replications is increased up to 10 or 20.

The tables from Table 3.33 to Table 3.35 represent the other performance values obtained to evaluate alternative policies.

Table 3.33 The Results Of The First Policy For The Assembly Department

Rep. No.	Factor Level 1							
	Time-in-System	The Uti. Of Trans.	The Uti. Of Feed. Op.	The Uti. Of Techn.	The Uti. Of 1st M. Sta.	The Uti. Of 2ndM. Sta.	The Uti. Of 3rdM. Sta.	The Uti. Of 4th M. Sta.
1	247.06	0.0996	0.1540	0.1819	0.8593	0.8221	0.7609	0.7409
2	280.87	0.0909	0.1406	0.2864	0.7867	0.7518	0.6997	0.6835
3	259.23	0.0984	0.1494	0.2149	0.8456	0.8058	0.7473	0.7302
4	267.41	0.0947	0.1430	0.2564	0.8189	0.7831	0.7308	0.7115
5	249.38	0.0943	0.1444	0.2787	0.8054	0.7683	0.7162	0.6992

Table 3.34 The Results Of The Second Policy For The Assembly Department

Rep. No.	Factor Level 2							
	Time-in-System	The Uti. Of Trans.	The Uti. Of Feed. Op.	The Uti. Of Techn.	The Uti. Of 1st M. Sta.	The Uti. Of 2ndM. Sta.	The Uti. Of 3rdM. Sta.	The Uti. Of 4th M. Sta.
1	280.56	0.2080	0.2942	0.5276	0.7517	0.7181	0.6694	0.6503
2	289.28	0.2075	0.2923	0.4798	0.7591	0.7251	0.6708	0.6516
3	264.54	0.2254	0.3168	0.4206	0.8265	0.7895	0.7386	0.7186
4	275.33	0.2147	0.3045	0.4431	0.8101	0.7731	0.7213	0.7009
5	289.52	0.2210	0.3053	0.4591	0.7952	0.7612	0.7032	0.6895

Table 3.35 The Results Of The Third Policy For The Assembly Department

Rep. No.	Factor Level 3							
	Time-in-System	The Uti. Of Trans.	The Uti. Of Feed. Op.	The Uti. Of Techn.	The Uti. Of 1st M. Sta.	The Uti. Of 2ndM. Sta.	The Uti. Of 3rdM. Sta.	The Uti. Of 4th M. Sta.
1	238.10	0.3483	0.4478	0.6642	0.7329	0.7612	0.6459	0.6287
2	377.15	0.3007	0.3808	0.7431	0.6471	0.6989	0.5729	0.5561
3	361.35	0.3403	0.4377	0.6572	0.7030	0.6186	0.6249	0.6074
4	335.60	0.3325	0.4207	0.6248	0.6587	0.6715	0.5846	0.5694
5	298.15	0.3054	0.3875	0.6719	0.6707	0.6277	0.5940	0.5810

The first column (time-in-system) in these tables represents the average total time to finish all of the assembly operations for making a V0. The columns from 2 to 8 represent the utilizations of the transporter, feeding operator, technician, the first, second, third and the fourth main assembly stations, respectively. It must be noted that the utilizations of the main assembly stations are equal to the utilization of any assembly machine in that main assembly station because the transfer mechanism (indexing tables) in the main assembly stations provides synchronous transfer of the base parts. As it was mentioned before, the utilization of the transporter in each policy is low in comparison to the utilizations of the technician and the feeding operator. Table 3.36 represents the 95% confidence intervals for these performance values.

Table 3.36 95% Confidence Intervals For The Performance Values

Factor Levels	Time-in-System	The Uti. Of Trans.	The Uti. Of Feed. Op.	The Uti. Of Techn.	The Uti. Of 1st M. Sta.	The Uti. Of 2ndM. Sta.	The Uti. Of 3rdM. Sta.	The Uti. Of 4th M. Sta.
1	243.59	0.091	0.140	0.189	0.787	0.751	0.701	0.684
	277.99	0.100	0.153	0.298	0.860	0.821	0.761	0.742
2	266.86	0.206	0.290	0.416	0.748	0.715	0.663	0.645
	292.83	0.225	0.315	0.517	0.829	0.719	0.739	0.720
3	252.63	0.299	0.378	0.618	0.639	0.610	0.567	0.552
	347.53	0.352	0.452	0.726	0.726	0.693	0.642	0.625

Based on Table 3.36, for example, 95% confidence interval of "Time-in-system" for the first policy is (243.59, 277.99). As it is seen in this table, the utilizations of the workers are statistically different in each policy (e.g., the confidence interval for the utilization of transporter in the first policy ,0.091-0.100, does not contain the lower or upper bounds of the confidence intervals for the second and the third policy). It must

be noted that the utilization of the transporter is very low in comparison to other workers. Therefore, as it was mentioned before, it can be assumed that the differences in performance values are not caused by the number of transporters in the assembly department. Except for the utilization of the second main assembly station, the utilizations of each main assembly station in the first and the second policy are not statistically different because the confidence intervals for the first policy contain the upper or the lower bound of the confidence intervals for the second policy. This conclusion supports the results obtained from Table 3.32.

Based on the results given above, it can be concluded that the second policy gives the best performance values in comparison to the first (current case) and the third policy. In the current case, 2 transporters, 6 feeding operators, and 6 technicians are employed for 6 assembly lines in the assembly department. The practical results obtained from this study suggest decreasing the number of technicians and the feeding operators from 6 to 3. This worker allocation policy is capable of meeting the current production target under the same setup conditions of the assembly lines.

3.4 Building The Model Of The Loading Department

As it was discussed before, the loading department consists of 4 identical lines each having 20 loading machines. Since these are identical lines, all of the necessary information to build the model of a loading line are valid for the other lines except for the distances between the pancake buffer and loading machines in each line. Therefore, in this section, only the input data and the logic employed in the first loading line will be explained step by step.

There are four types of resources in a loading line. These are loading machines (20 units), conveyors (2 units), operator for feeding the pancakes (1 unit), and operator for feeding VOs (1 unit). V0 feeding operator feeds the first conveyor with VOs. The VOs which are loaded onto the first conveyor are transferred to the first available loading machine. The statement “Available Loading Machine” implies the loading machine of which the current queue capacity is greater than zero. When the V0 is

transferred to the first available loading machine, the V0 is removed from the first conveyor and is placed into the waiting area of this selected machine. By using the pancakes, the loading machine loads the V0s with video tape in a specified length. When the operation is finished, the V0 is removed from the loading machine and is sent to the second conveyor. If an available machine can not be found for a V0 through the line, it is kept on the first conveyor until the last loading machine in the line becomes available. At this time, the first conveyor and the V0 feeding operation stops. When the length of the video tape in a pancake equals to zero, the pancake feeding operator is requested to feed the loading machine with a pancake.

The aim of developing the model of the loading department is to evaluate the effects of changing the number of pancake feeding operators on daily throughput of the first line, time-in-system, and the utilization of the pancake feeding operator. Currently, the policy of “1 operator per line” is applied. Following the model building, verifying and validating, an experiment plan is designed. The effects of the number of pancake feeding operators employed per line is evaluated in three levels:

- One pancake feeding operator per line (Four operators per department).
- One pancake feeding operator per two lines (Two operators per department).
- One pancake feeding operator per four lines (One operator per department).

As it was mentioned above, the performance measures to compare these alternative policies are “the daily throughput of the first loading line”, and “the utilization of the pancake feeding operator”. “Time-in-system for a V0 in the first line” is also evaluated as a secondary measure to support the analysis.

3.4.1 Assumptions Of The Model

The assumptions made in building the model of the loading department can be listed as follows:

- The V0s are introduced to the system in the batch size of 1.

- Since the V0 feeding operation is quite simple, the time required for this operation is assumed to be constant. This means that the first conveyor is fed with V0s at a constant rate. The feeding rate does not change between the shifts (3-shifts).
- Pancakes are transferred from the warehouse to the loading department continuously. Therefore, arrivals of the pancakes have been modeled as if there were infinite pancake buffer in the loading department.
- The operator feeding pancakes can carry only one pancake and each time he is requested, he must go to the pancake buffer to take one.
- The layout of the pancake buffer (i.e., pancake buffer serves as a pancake storage) can not be changed between the different policies.
- Since each line has 20 parallel loading machines which are dedicated to the same operation, a failure in any loading machine does not seriously affect the whole line.
- Each line was setup according to the only one type of video tape length.

3.4.2 Input Data Of The Model

Based on the assumptions given above, two groups of data are needed to be collected to build the model. The first group of data has deterministic nature. Deterministic data group can be summarized as follows:

- The batch size of V0.
- The time between arrivals of V0.
- The buffer capacity of the loading machines.
- The speed of the conveyors.
- The speed of the pancake feeding operator.
- Operation time of the loading machines.
- Video tape length to be loaded (For a 3-hour video cassette).
- Video tape length in a pancake.
- Distances between the pancake buffer and the loading machines (For Pancake

Feeding Operator).

- Lengths of the conveyor segments.

According to the first assumption, at the beginning of each line, the VOs are introduced into the system in the batch size of 1. This logic avoids the large amount of queues, and saves in computational time. The time between arrivals is constant and equal to 1.5 seconds as it was explained in the second assumption. Each loading machine has its own buffer with the capacity of 15 VOs. As it was mentioned before, there are two conveyors at each line. While the first one transfers the VOs to the loading machines, the second one transfers the loaded VOs to the packaging department. The speed of each conveyor in the loading department is constant (constant line speed) and equals to 5 cassette length units/second. The type of the conveyors is accumulating. In the case of accumulating conveyor, the entity accumulation on the conveyor between the stations is allowed. The speed of pancake feeding operator is 10 cassette length units/second.

Since the loading machines are highly automated, the time to load a VO with video tape in a specified length is constant and the time to carry out this operation equals to 37 seconds. This duration is valid for only a 3-hour video cassette. The video tape length in a 3-hour video cassette is 250 meters. Pancakes are bought from abroad in a standard form and length. The video tape length in a pancake is 12.000 meters. The distances between the pancake buffer and the loading machines for a pancake feeding operator can be seen in Table 3.37.

In Table 3.37, while the column which is named “No.” shows the loading machines in the related loading line, the column which is named “Dist.” shows the distances between the pancake buffer and related loading machine in unit of video cassette length. For example, the distance between the pancake buffer and the first loading machine of second loading line is 15 cassette length units. The last deterministic data group in the model is the length of the segments of the conveyors. The segment length between two stations is 7 cassette length units. This is valid for both the first and the second conveyor.

**Table 3.37 The Distances Between The Pancake Buffer and
The Loading Machines**

Loading Line 1				Loading Line 2				Loading Line 3				Loading Line 4			
No.	Dist.	No.	Dist.	No.	Dist.	No.	Dist.	No.	Dist.	No.	Dist.	No.	Dist.	No.	Dist.
1	5	11	55	1	15	11	65	1	25	11	75	1	35	11	85
2	10	12	60	2	20	12	70	2	30	12	80	2	40	12	90
3	15	13	65	3	25	13	75	3	35	13	85	3	45	13	95
4	20	14	70	4	30	14	80	4	40	14	90	4	50	14	100
5	25	15	75	5	35	15	85	5	45	15	95	5	55	15	105
6	30	16	80	6	40	16	90	6	50	16	100	6	60	16	110
7	35	17	85	7	45	17	95	7	55	17	105	7	65	17	115
8	40	18	90	8	50	18	100	8	60	18	110	8	70	18	120
9	45	19	95	9	55	19	105	9	65	19	115	9	75	19	125
10	50	20	100	10	60	20	110	10	70	20	120	10	80	20	130

The only stochastic component of the system is the time to feed the loading machines with pancake. Firstly, to find out which distribution represents best this stochastic component, 17 observations were taken randomly in two days with a digital (1/100 sec.) chronometer. Table 3.38 shows these observations and the summary results of the analysis. Based on this table, it is decided that the time to pancake feeding operation is normally distributed with the mean and the standard deviation of 54.8 and 5.11, respectively.

3.4.3 Definition Of The Attributes, Variables, And Elements

In this section, the attributes, the variables, and the elements of the model will be defined. The term “L1” at the beginning of each definition implies the loading line number. This term takes the form of “L2, L3, and L4” for the second, third and fourth loading lines, respectively. The attributes, variables, and elements used in this model can be listed as follows:

- **i:** It is a control attribute and stores the loading station number for the first loading line. It takes values from 1 to 20. This control attribute is “k”, “l”, and “t” for the second, third and the fourth loading line, respectively.

**Table 3.38 Curve Fit Summary For The Operation of Feeding Pancake
In The Loading Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	48.97	10	47.36
2	52.68	11	54.30
3	57.32	12	51.16
4	58.79	13	57.42
5	56.57	14	53.79
6	59.85	15	54.43
7	44.35	16	64.02
8	63.18	17	56.34
9	51.38	18	-

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	2	48.2	0.118	0.081	0.118	0.098
2	3	52.4	0.176	0.220	0.294	0.318
3	6	56.6	0.353	0.318	0.647	0.636
4	4	60.8	0.235	0.243	0.882	0.879
5	2	65.0	0.118	0.098	1.000	0.977

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	panfeed.txt	Function	Square Error
Histogram Range	44 to 65	Normal	0.00496
No. of Data Points	17	Triangular	0.005564
No. of Intervals	5	Beta	0.01449
Min. Data Value	44.4	Weibull	0.01662
Max. Data Value	64	Gamma	0.03701
Sample Mean	54.8	Uniform	0.03875
Sample Std. Deviation	5.27	Erlang	0.04327
Distribution Function	Normal	Lognormal	0.07181
Siman Usage	NORM(54.8,5.11)	Exponential	0.1061

- **Lineno:** It is a control attribute and stores the loading line number. It takes values from 1 to 4 (four loading line).
- **Timein1:** It is a control attribute and stores the time when the VOs are introduced into the system. For the other loading lines, it is referred as “Timein2”, “Timein3”, and “Timein4”.
- **L1pan(i):** It is a variable and represents the video tape length (in meter) in the pancake in i_{th} loading machine. Its value is decreased by 250 meters after each loading operation. It has the initial value of 12.000 meters.
- **L1batch:** It is a variable and represents the batch size of arrivals. Its initial value is 1. If an available loading machine can not be found for a VO through the line, a value of 0 is assigned to “L1batch” until the last loading machine is available.
- **L1enter:** The name of the station where the VOs are introduced into the system.
- **L1end:** The name of the station where the loaded VOs are transferred by the second conveyor.
- **L1pancake:** The name of the station which is at the beginning of the first loading line. This station has infinite pancake buffer and the pancake requests from each line are met from this unique station.
- **L1con1q:** The name of the queue for conveyor “L1con1”.
- **L1con1:** This is the name of the first conveyor. It transfers the VOs from the station “L1enter” to an available loading machine.
- **L1con2q:** The name of the queue for conveyor “L1con2”.
- **L1con2:** This is the name of the second conveyor. It transfers the loaded VOs from the loading station to station “L1end”.
- **L1con1seg:** The name of the segment set of conveyor “L1con1”.
- **L1con2seg:** The name of the segment set of conveyor “L1con2”.
- **tr1q:** The name of the queue for the pancake feeding operator “tr1”.
- **tr1:** The name of the pancake feeding operator.
- **tr1dis:** The name of the distance set which is given in Table 3.37.
- **L1s(i):** The name of the station set which shows the i_{th} member of that set.
- **L1s1, .., L1s20:** The members of set “L1s”.
- **L1r(i):** The name of the loading machine set which shows the i_{th} member of

that set.

- **L1r1, .., L1r20:** The members of set “L1r”.
- **L1q(i):** The name of the queue of the loading machine set which shows the i_{th} member of that set.
- **L1q1, .., L1q20:** The members of set “L1q”.
- **L1jobs:** It is a counter and it stores the total number of the loaded VOs.

Based on the explanations made so far, the general entity flow in the loading department for a line can be summarized in Figure 3.10. The SIMAN model and experiment listing files for the loading department can be seen in Appendix B.1 and Appendix B.2, respectively.

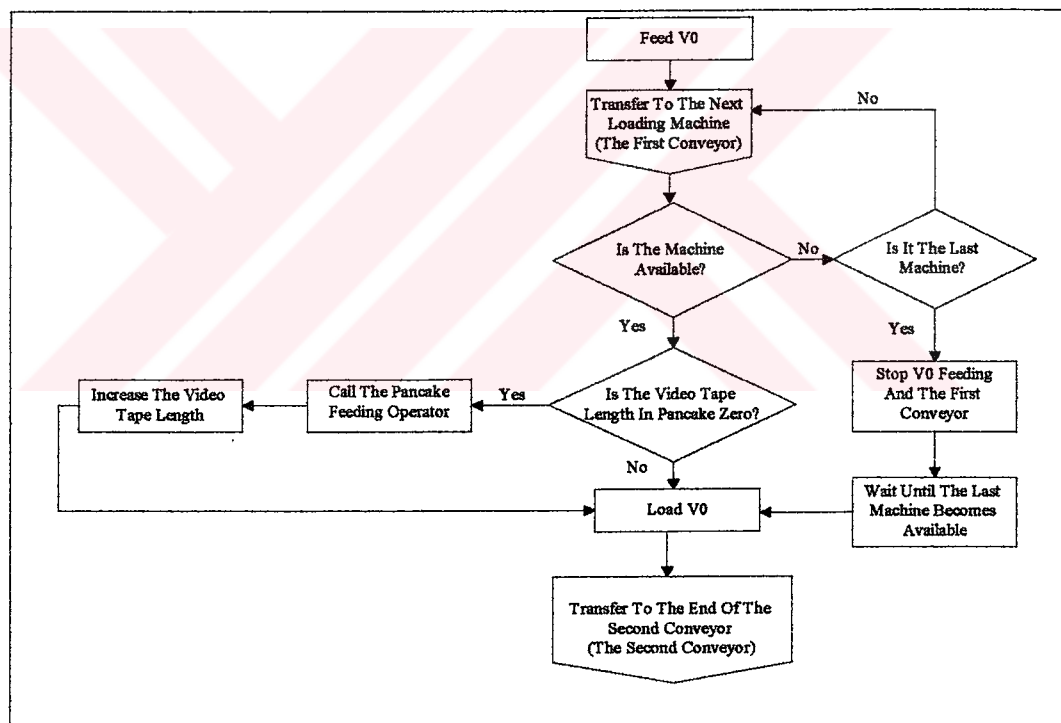


Figure 3.10 The General Algorithm For A Loading Line

3.4.4 Validating The Model Of A Loading Line

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Since the loading department operates continuously for 24 hours, the methodology of analyzing nonterminating systems needs to be used. The pilot run has been made to

decide on the truncation point. In the validation process, the first loading line is evaluated for daily throughput. As it was mentioned in Section 3.3.5, the statistics to be collected for generating a time plot graph is “time-in-system”, and the selected replication length of the pilot run is 86400 seconds (24 hours). During the pilot run, the flowtime (time-in-system) of each V0 is saved in a file. Based on the results of the pilot run, truncation point of the system is selected as $t=4000$ seconds (total observations=44166). Since the performance measure of the validation process is the daily throughput of the first line, 10 independent replications with the length of the $86400+4000=90400$ seconds were made. The 10 independent daily throughputs of the first loading line can be seen in Table 3.39.

Table 3.39 Simulation Results Of Ten Replications For A Loading Line

Replication No.	Loaded V0s	Replication No.	Loaded V0s
1	44427	6	44251
2	44269	7	44403
3	44194	8	44293
4	44139	9	44208
5	44244	10	44304

Since the loading lines are highly automated, the daily or hourly throughput can be monitored. Each machine has its own digital counter, and it is possible to see the throughput of each loading machine in any time interval. The long period-analysis shows that the first loading line has a capability of loading 44250 V0s per day for current setup conditions. This expectation can be compared with the results obtained from the simulation output to prove that the model is valid. Since these are independent observations t-test can be applied. The hypotheses to be tested has been expressed as follows:

$$H_0 : \mu=44250 \text{ Loaded V0s per the first line}$$

$$H_1 : \mu \neq 44250$$

$$\text{Critical t-value for } \alpha=0.05 \text{ and the degree of freedom D.F}=9 \Rightarrow t_{n-1, 1-\alpha/2}=2.262$$

$$x = 44273 \text{ loaded V0s per the first line}$$

$$S_{\bar{x}} = 28.233$$

$$t\text{-statistics} \Rightarrow (\bar{x} - \mu) / S_{\bar{x}} = 0.822 \Rightarrow 0.822 < 2.262 \quad H_0 \text{ is accepted.}$$

Since the H_0 hypothesis is accepted, it can be said that the mean of the simulation output for daily throughput is not different from the expected value in %95 confidence level.

3.4.5 The Results Of Experimental Studies For The Loading Department

Following the model validation, the last step in this analysis is to design an experiment plan and to analyze the results of the experiments. As it was mentioned before, the objective of simulating the loading department is to determine the effects of changing the number of pancake feeding operators on following performance measures:

- The daily throughput of the first loading line.
- The utilization of the pancake feeding operator.
- Time-in-system for a V0 in the first loading line.

Table 3.40 summarizes the factor plan which was adopted.

Table 3.40 The Factor Plan For The Loading Department

Factor	Factor Levels		
	1	2	3
The number of pancake feeding operator in the loading department	1 operator per line (4 operators in the department totally)	1 operator per 2 lines (2 operators in the department totally)	1 operator per 4 lines (1 operator in the department totally)

The input data to run these experiments were explained in Section 3.4.2 Since the objective is to observe the differences between the steady-state behaviours of these policies, two additional pilot runs were made to identify the truncation point of the factor levels 2 and 3. These were identified as it was explained in Section 3.3.5. Table 3.41 summarizes the replication conditions for three factor levels.

Table 3.41 The Replication Conditions For The Loading Department

Conditions	Factor Levels		
	1	2	3
Warm-up Period	4000	7500	6000
Replication Length	90400	93900	92400
No. of Replications	10	10	10
Initialize System	yes	yes	yes
Initialize Statistics	yes	yes	yes

Based on these experimental conditions, 10 independent replications were made and the simulation results summarized in Table 3.42 were obtained.

Table 3.42 The Summary Results For The Loading Department

Rep. No.	Factor Levels								
	1			2			3		
	Thr.	Uti.	Time-in-Sys.	Thr.	Uti.	Time-in-Sys.	Thr.	Uti.	Time-in-Sys.
1	44427	0.7001	572.50	31311	0.9990	683.18	15176	0.9959	1437.0
2	44269	0.6961	574.73	31343	0.9988	685.19	15184	0.9956	1540.7
3	44194	0.6954	574.23	31281	0.9984	684.29	15177	0.9961	1502.9
4	44139	0.6955	573.17	31235	0.9992	683.35	15179	0.9959	1397.4
5	44244	0.6977	573.60	31308	0.9998	684.77	15180	0.9960	1405.0
6	44251	0.6995	573.54	31296	0.9993	682.00	15180	0.9960	1440.9
7	44403	0.6981	571.94	31416	0.9992	681.81	15185	0.9959	1547.9
8	44293	0.6934	573.84	31464	0.9996	681.15	15186	0.9952	1323.2
9	44208	0.6954	574.27	31155	0.9988	685.86	15183	0.9958	1313.5
10	44304	0.6958	574.07	31391	0.9979	684.32	15177	0.9960	1470.4

As it is seen in Table 3.42, the changes in the number of pancake feeding operators in the loading department have a great effect on daily throughput of the first loading line. Under the first policy, the average daily throughput of the first line is 44250 loaded VOs, and one operator is in charge of feeding 20 loading machines. On the other hand, under the second policy, one operator is in charge of feeding 40 loading machines and while this causes an increase in operator utilization, it leads to a big amount of decrease in daily throughput of the first line. This means that the loading machines stay in “wait state” for a long time in the second policy. These results show that the first loading line is very sensitive to the changes in the number of operators.

Based on this analysis, it can be concluded that the current policy is the best one in comparison to the other policies for a given production target and the setup

conditions of the line such as, conveyor speed, V0 feeding rate, and product type. This environment will enable the production managers to figure out how to allocate the work force levels to meet the planned production targets. If the video cassette demand fluctuates, this model can be used to evaluate different work force allocation strategies according to the predefined production level.

3.5 Building The Model Of The Packaging Department

Following the loading operation, the loaded V0s are transferred to the packaging department via the second conveyor of the loading line. The packaging department consists of 4 packaging lines each having 2 parts. The first part of a packaging line has automated operations and the other part of a packaging line has manual operations. All of the lines in the packaging department are identical.

There are four types of resources in a packaging line. These are automated packaging machines (8 units), casing operators (2 units), large casing operator (1 unit) and conveyor (1 unit). The conveyor of a packaging line transfers the loaded V0s from the loading line to the automated packaging machines. Since the speed of a loading line is greater than a packaging line, the worker removes the loaded V0s from the conveyor as soon as they start accumulating on the conveyor. This process which takes place just at the beginning of the packaging line provides independency between the loading lines and the packaging lines. The automated machines prepare the loaded V0s for casing and large casing operations. At the end of each automated part of the packaging line, two casing operators who are dedicated to the same operation, make a case, then place 10 video cassettes into that case. Following this operation, the large casing operator makes a large case, then places 5 cases into that large case (i.e., one large case consists of 50 video cassettes, one case consists of 10 video cassettes). During the plant trips, it has been observed that a queue has been occurring at the end of the automated part of the packaging lines. This indicates that the speed of the manual part is not fast enough to meet the capacity of the automated part. Furthermore, the capacity of this queue is limited due to the layout conditions. As soon as the capacity of the queue reaches a predefined level, the automated part of the

packaging line is stopped until the number of video cassettes in the queue is equal to zero.

After the model building, an experiment plan is designed to compare the alternative work force allocation policies with the current policy. The two factors which are studied in different levels are the number of casing operators and the number of large casing operators in the packaging department. Since two stage experimentation provides less number of replications, these factors are evaluated in two stages:

- Evaluate the effects of changing the number of casing operators per line. Choose the design which gives the best performance measure.
- According to the conclusion obtained from the first stage, evaluate the effects of changing the number of large casing operators per line.

The first factor (the number of casing operators per line) is evaluated in two levels:

- Current policy, 2 casing operators per line ($4 \times 2 = 8$ operators)
- 3 casing operators per line ($4 \times 3 = 12$ operators)

Based on the best design configuration obtained in the first stage, the second factor (the number of large casing operators per line) is evaluated in three levels:

- 1 large casing operator per line ($4 \times 1 = 4$ operators)
- 1 large casing operator per 2 lines ($2 \times 1 = 2$ operators)
- 1 large casing operator per 4 lines ($1 \times 1 = 1$ operators)

The objective in developing the model of the packaging lines is to determine the number of casing and large casing operators to be employed in the packaging lines so that the manual operations will not be bottleneck in meeting the daily production targets. The performance measures used to evaluate the effects of different policies are throughput of the department, time-in-system, and the utilization of the operators.

3.5.1 Assumptions Of The Model

The assumptions made in building the model of the packaging department can be listed as follows:

- The packaging department is independent from the loading department.
- Loaded VOs are introduced into the system at a constant rate with a batch size of one. This rate does not change between the shifts.
- The components used in automated packaging operations are available when requested (i.e., infinite component buffer for each automated machine).
- Automated machines are available continuously (No failure).
- Each line was setup according to the only one type of case and large case combination (A case consists of 10 video cassettes and a large case consists of 5 cases).

3.5.2 Input Data Of The Model

Based on the assumptions given above, two groups of data are needed to be collected to build the model. The first group of data has deterministic nature. Deterministic data group can be summarized as follows:

- The batch size of loaded VOs.
- The time between arrivals of loaded VOs.
- The queue capacity for casing operation.
- The speed of the conveyor.
- Operation time of the automated packaging machines.
- Lengths of the conveyor segments.

Based on the first assumption given above, the batch size of loaded VOs is 1. The time between arrivals are constant and equal to 1.5 seconds. As it was mentioned before, there are queues with the capacity of 100 loaded VOs (It was modeled as a

case unit, i.e., capacity is $100/10=10$) at the beginning of each casing operation. The speed of the conveyor is 5 cassette length units/second. The automated part of a packaging line consists of 8 serial machines which are dedicated to the different operations on the loaded VOs. Table 3.43 represents the operation times of these machines in seconds (1/100 second, digital chronometer). These values have been obtained from the factory records.

Table 3.43 The Operation Times Of The Packaging Machines

Machine No.	1	2	3	4	5	6	7	8
Operation Time	1.69	1.75	1.65	1.80	1.61	1.73	1.54	1.77

The segment length between two automated packaging stations are 7 cassette length units. The queue selection rule used in this model is cyclic rule. In this rule, jobs select the queues in cyclic order. For example, the first 10 video cassettes select the first casing operator, the second 10 video cassettes select the second casing operator, then the third 10 video cassettes select the first casing operator again. This rule provides approximately equal work load to each casing operator in the same packaging line.

To find out which distribution represents best the stochastic nature of casing and large casing operations, 20 observations have been taken for each operation. Table 3.44 and Table 3.45 indicate the observation values and the summary results of the ARENA "Input Analyzer".

Based on Table 3.44, the selected distribution for the casing operation is normal distribution with a mean and standard deviation of 41.9, 4.23, respectively. With respect to Table 3.45, it is decided that the time for large casing operation is normally distributed with a mean and standard deviation of 39.1, 6.06, respectively. Since the packaging lines are identical, all of the data mentioned in this section are valid for the other three lines.

**Table 3.44 Curve Fit Summary For The Operation of *Casing*
In The Packaging Department**

Observation No.		Obs. Value (Sec.)		Observation No.		Obs. Value (Sec.)	
1		45.57		11		36.77	
2		44.91		12		40.55	
3		37.49		13		43.18	
4		45.67		14		39.91	
5		42.41		15		44.48	
6		42.17		16		49.12	
7		43.42		17		42.21	
8		48.36		18		35.05	
9		37.74		19		47.18	
10		34.12		20		38.50	

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	3	37.2	0.150	0.101	0.150	0.131
2	4	40.4	0.200	0.227	0.350	0.358
3	6	43.6	0.300	0.295	0.650	0.653
4	4	46.8	0.200	0.222	0.850	0.875
5	3	50.0	0.150	0.097	1.000	0.972

SUMMARY REPORT			BEST FIT SUMMARY	
Data File	casing.txt		Function	Square Error
Histogram Range	34 to 50		Normal	0.006467
No. of Data Points	20		Beta	0.007647
No. of Intervals	5		Uniform	0.015
Min. Data Value	34.1		Weibull	0.01556
Max. Data Value	49.1		Triangular	0.0166
Sample Mean	41.9		Erlang	0.0253
Sample Std. Deviation	4.34		Gamma	0.0311
Distribution Function	Normal		Lognormal	0.06912
Siman Usage	NORM(41.9,4.23)		Exponential	0.07377

**Table 3.45 Curve Fit Summary For The Operation of Large Casing
In The Packaging Department**

Observation No.	Obs. Value (Sec.)	Observation No.	Obs. Value (Sec.)
1	32.76	11	38.36
2	37.22	12	45.40
3	37.99	13	42.91
4	38.55	14	38.54
5	41.89	15	39.31
6	36.22	16	50.83
7	27.23	17	41.61
8	49.81	18	32.07
9	35.66	19	48.24
10	30.83	20	36.67

Int. No.	No. Of Data Pts.	Upper Bound	Probability Density		Cumulative Distribution	
			Data	Function	Data	Function
1	2	31.80	0.100	0.091	0.100	0.113
2	4	36.60	0.200	0.226	0.300	0.338
3	7	41.40	0.350	0.308	0.650	0.646
4	4	46.20	0.200	0.232	0.850	0.878
5	3	51.00	0.150	0.096	1.000	0.974

SUMMARY REPORT		BEST FIT SUMMARY	
Data File	lcasing.txt	Function	Square Error
Histogram Range	27 to 51	Normal	0.006425
No. of Data Points	20	Triangular	0.008486
No. of Intervals	5	Beta	0.01639
Min. Data Value	27.2	Weibull	0.01823
Max. Data Value	50.8	Uniform	0.035
Sample Mean	39.1	Gamma	0.03708
Sample Std. Deviation	6.22	Erlang	0.04104
Distribution Function	Normal	Lognormal	0.07792
Siman Usage	NORM(39.1,6.06)	Exponential	0.1098

3.5.3 Definition Of The Attributes, Variables, And Elements

In this section, the attributes, the variables, and the elements of the model will be defined. The term “L1” at the beginning of each definition implies the packaging line number. This term takes the form of “L2, L3, and L4” for the second, third and fourth packaging lines, respectively. The attributes, variables, and elements used in this model can be listed as follows:

- **i:** It is a control attribute and stores the packaging station number for the automated part of the first packaging line. It takes values from 1 to 8. This control attribute is “k”, “l”, and “n” for the second, third and the fourth packaging lines, respectively.
- **Timein1:** It is a control attribute and stores the time that the loaded VOs are introduced into the system.
- **L1pan(i):** It is a variable and represents the operation time of i_{th} packaging machine.
- **L1batchsize:** It is a variable and represents the batch size of arrivals. Its initial value is 1. If the number of loaded VOs waiting for the conveyor is equal to 10, the value of 0 is assigned to “L1batchsize” until the number of loaded VOs waiting for the conveyor is equal to 0.
- **L1enter:** The name of the station where the loaded VOs are introduced into the system.
- **L1end:** The name of the station which is at the end of the automated part of a packaging line.
- **L1con1q:** The name of the queue for conveyor “L1con1”.
- **L1con1:** This is the name of the conveyor. It transfers the loaded VOs from the station “L1enter” to the station “L1end” through the line.
- **L1con1seg:** The name of the segment set of conveyor “L1con1”.
- **L1s(i):** The name of the station set which shows the i_{th} member of that set.
- **L1s1, .., L1s8:** The members of set “L1s”.
- **L1r(i):** The name of the packaging machine set which shows the i_{th} member of

that set.

- **L1r1, .. , L1r8:** The members of set “L1r”.
- **L1q(i):** The name of the queue of the packaging machine set which shows the i_{th} member of that set.
- **L1q1, .. , L1q8:** The members of set “L1q”.
- **L1m1r, .. , l1m3r:** The name of the casing operators.
- **L1m1q, .. , l1m3q:** The name of the queues for casing operators.
- **L1m4r:** The name of the large casing operator.
- **L1m4q:** The name of the queue for large casing operator.
- **L1jobs:** It is a counter and it stores the total number of large cases.

Two different models were established to analyze the packaging department. The first model includes only the first packaging line with two casing operators and one large casing operator. This model was used to analyze the current policy. Based on the results (i.e., two or three casing operators per line) obtained from the first model, the second model was established. The second model includes four packaging lines each having 2 or 3 casing operators (i.e., The policy depends on the results of the first model), and one large casing operator. This model can be used as a test bed to evaluate the policy of “1 large case operator per line”, “1 large case operator per 2 lines”, and “1 large case operator per 4 lines”. For example, if the initial value of zero is assigned to L2batchsize, L3batchsize, L4batchsize and the initial value of 1 is assigned to L1batchsize, only the first line operates under the policy of “1 large case operator per line”. In this case, counter “L1jobs” stores the total number of large cases for one line. If the initial value of zero is assigned to, L3batchsize, L4batchsize and the initial value of 1 is assigned to L1batchsize, L2batchsize, only the first and the second line operate under the policy of “1 large case operator per two lines”. In this case the counter “L1jobs” stores the total number of large cases for two lines.

Based on the explanations made so far, the general entity flow in the packaging department for a line can be summarized in Figure 3.11. The SIMAN model and experiment listing files for the packaging department can be seen in Appendix C.1 and

Appendix C.2, respectively.

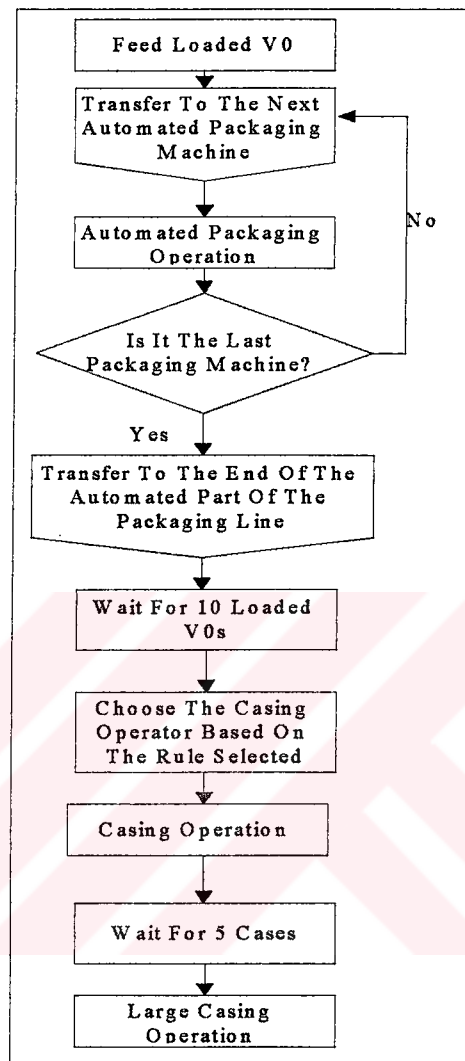


Figure 3.11 The General Algorithm For A Packaging Line

3.5.4 Validating The Model Of A Packaging Line

The current case was evaluated to validate the first model. In the current policy (i.e., two casing operators and one large casing operator per line), it is known that the first packaging line operates with the capacity of 825 large cases per day. This value can be compared with the results of ten replications. The replication conditions were determined according to the Moving Average Module of the ARENA -Output Analyzer-. Time-in-system was collected during the pilot run. According to the pilot

run, the truncation point was selected as 12000 seconds. The replication length was $12000+86400=98400$ seconds and the number of replications is 10. Table 3.46 represents the results of ten replications.

Table 3.46 The Simulation Results Of Ten Replications For The First Packaging Line

Replication No.	No. of L. Cases	Replication No.	No. of L. Cases
1	824	6	826
2	825	7	827
3	824	8	825
4	826	9	825
5	826	10	824

The hypotheses to be tested has been expressed as follows:

$H_0 : \mu=825$ Large Cases per the first line

$H_1 : \mu \neq 825$

Critical t-value for $\alpha=0.05$ and the degree of freedom D.F=9 $\Rightarrow t_{n-1, 1-\alpha/2}=2.262$

$\bar{x} = 825.2$ Large Cases per the first line

$S_{\bar{x}} = 0.3266$

t-statistics $\Rightarrow (\bar{x} - \mu) / S_{\bar{x}} = 0.612 \Rightarrow 0.612 < 2.262$ H_0 is accepted.

Since the H_0 hypothesis is accepted, it can be said that the mean of the simulation output for daily throughput is not different from the expected value in %95 confidence level.

3.5.5 The Results Of Experimental Studies For The Packaging Department

Before the experimentation, it is useful to determine several limits on the daily throughput of a packaging line. As it was mentioned before, expected daily throughput of the first loading line is 44250 loaded VOs. This means that the capacity of the first packaging line needs to be $44250/50=885$ large cases. However, in the

current case, the daily throughput is 825 large cases. The differences between these two numbers give us the number of loaded VOs to be removed from the conveyor (i.e., $885-825=60$ large cases, $60*50=3000$ loaded VOs).

It is known that the speed of the first loading line is greater than the speed of the automated part of the first packaging line. Therefore, the maximum capacity of the automated part of the first packaging line needs to be determined. This can be achieved by assigning the value of zero to the operation times of all manual operations. One replication is enough to understand this deterministic behaviour. After executing the model for zero time delay (i.e., casing and large casing operation times are equal to zero), the maximum daily throughputs of 864 large cases were obtained. This number can be useful to evaluate the effects of the number of casing operators per packaging line. If the alternative design (i.e., 2 or 3,..., casing operators per line) provides daily throughput of 864 large cases, additional operator has no effects on daily throughput but causes a decrease in operator utilization.

Following the model building, verifying and validating, an experiment plan is designed to compare the alternative policies with the current policy. Two factors are evaluated in two stages. Table 3.47 summarizes the factor plan which was applied.

Table 3.47 The Factor Plan For The Packaging Department

Factor	Factor Levels		
	1	2	3
The number of casing operators in the packaging department	2 operators per line (8 operators in the department totally)	3 operators per line (12 operators in the department totally)	-
The number of large casing operator in the packaging department	1 operator per line (4 operators in the department totally)	1 operator per 2 lines (2 operators in the department totally)	1 operator per 4 lines (1 operator in the department totally)

The performance measures to compare these alternative designs are, daily throughput of the packaging department, time-in-system, and the utilization of the operators.

Table 3.48 represents the replication conditions for evaluating the number of casing operators per line (the first stage of the experimentation). The length of the warm up period was determined according to the moving average analysis.

Table 3.48 The Replication Conditions For The First Stage Of Experiment

Conditions	Factor Levels	
	1	2
Warm-up Period	12000	1500
Replication Length	98400	87900
No. of Replications	10	10
Initialize System	yes	yes
Initialize Statistics	yes	yes

Factor level “1” represents the policy of “two casing and one large casing operators per line”. On the other hand, factor level “2” represents the policy of “three casing and one large casing operators per line”. The simulation results which were obtained under these experimental conditions are summarized in Table 3.49.

Table 3.49 Summary Results For The First Policy In The Packaging Dept.

Rep. No.	Factor Level				
	1				
	Daily Throughput	Uti. of 1th Casing O.	Uti. of 2nd Casing O.	Uti. of Large Casing O.	Time-in-System (Sec.)
1	824	1.0000	0.9967	0.3728	561.27
2	825	1.0000	0.9989	0.3720	543.42
3	824	1.0000	0.9964	0.3655	552.78
4	826	1.0000	1.0000	0.3739	613.66
5	826	0.9998	1.0000	0.3741	641.92
6	826	1.0000	0.9993	0.3744	594.33
7	827	1.0000	1.0000	0.3742	663.82
8	825	1.0000	0.9997	0.3757	573.01
9	825	1.0000	1.0000	0.3733	601.98
10	824	1.0000	1.0000	0.3747	610.73

According to Table 3.49, it is clear that two casing operators per line can not meet the capacity requirements of the automated part of the packaging line. The average throughput under this policy is approximately 825 large cases. As it is calculated before, the maximum capacity of the automated part is 864 large cases. This difference ($864 - 825 = 39$ large cases) shows that additional casing operator may

increase the daily throughput. The high utilizations of the casing operators (i.e., approximately 100%) indicate that the manual operations are the bottleneck of the packaging line. As it was mentioned before, the queues occurring at the end of the automated part result in the line stops which decrease the daily throughput of the line. On the other hand, the utilization of large casing operator is approximately 37%. This low level of utilization implies that the large casing operator could be used for two lines. Table 3.50 represents the summary results for the factor level “2” (i.e., three casing and one large casing operators per line).

Table 3.50 Summary Results For The Second Policy In The Packaging Dept.

Rep. No.	Factor Level					
	2					
	Thr.	Uti. of 1th Casing O.	Uti. of 2nd Casing O.	Uti. of 3rd Casing O.	Uti. of Large Casing O.	Time-in-Sys. (Sec.)
1	864	0.6995	0.6960	0.7010	0.3903	230.60
2	863	0.6978	0.6988	0.6990	0.3887	230.76
3	864	0.6986	0.6983	0.6970	0.3856	230.26
4	863	0.6998	0.6964	0.6980	0.3869	230.62
5	864	0.6959	0.6970	0.6996	0.3875	230.50
6	864	0.6977	0.6976	0.7001	0.3888	230.62
7	864	0.6954	0.6994	0.6978	0.3852	230.85
8	864	0.6955	0.6970	0.7008	0.3873	230.94
9	863	0.6986	0.6998	0.6995	0.3891	230.63
10	864	0.6947	0.7010	0.6961	0.3881	230.64

According to the Table 3.50, it can be said that second factor level gives the better results in comparison to the first factor level. In this case, the daily throughput of the line increases because of the decreases in line stops. Since there are no queues near the casing operators, the automated part of the packaging line does not stop. The daily throughput of the packaging line under the second policy is almost equal to the maximum capacity of the line which was calculated (864 large cases) earlier. The utilizations of the casing operators approximately decreases from 100% to 70% in comparison to the first factor level. On the other hand, the utilization of large casing operator increased from 37% to 39%. The decrease in the “time-in-system” shows that the second policy causes the entities spend less time in the system. Based on the results of the first stage, it was decided that the number of casing operators to be used

per line should be three.

In the second stage of the experimentation the factor to be evaluated is the number of large casing operator in the packaging department under the assumption of “three casing operators per line”. This factor has been evaluated in three levels:

- 1 large casing operator per line ($4 \times 1 = 4$ operators)
- 1 large casing operator per 2 lines ($2 \times 1 = 2$ operators)
- 1 large casing operator per 4 lines ($1 \times 1 = 1$ operators)

These policies are evaluated according to the performance measures of “the daily throughput of the department”, “the utilization of the large casing operator”, and “time-in-system”.

At the beginning of the study, a pilot run for each policy was made to understand the steady state behaviour of the policies. Moving Average Module was used to identify the truncation points. The first policy has been already evaluated in the first stage of experimentation. The results of the pilot run for the third policy indicated that the system could not reach a steady state condition. Therefore, no truncation point has been identified. Table 3.51 represents the starting conditions for selected factor levels.

Table 3.51 The Replication Conditions For The Second Stage Of Experiment

Conditions	Factor Levels		
	1	2	3
Warm-up Period	1500	750	0
Replication Length	87900	87150	86400
No. of Replications	10	10	0
Initialize System	yes	yes	yes
Initialize Statistics	yes	yes	yes

The results of simulation summarized in Table 3.52. This table indicates that the second policy gives the best performance measure in comparison to the other two policies.

Table 3.52 The Summary Results For The Packaging Department

Rep. No.	Factor Levels								
	1			2			3		
	Thr.	Uti.	Time-in-Sys.	Thr.	Uti.	Time-in-Sys.	Thr.	Uti.	Time-in-Sys.
1	3456	0.3903	230.60	3456	0.7793	191.16	2217	0.9988	15531
2	3452	0.3887	230.76	3456	0.7747	190.88	2202	0.9988	15656
3	3456	0.3856	230.26	3456	0.7793	191.31	2197	0.9988	15811
4	3452	0.3869	230.62	3456	0.7852	191.46	2203	0.9989	15795
5	3456	0.3875	230.50	3456	0.7819	191.52	2208	0.9987	15677
6	3456	0.3888	230.62	3456	0.7856	191.57	2207	0.9988	15751
7	3456	0.3852	230.85	3456	0.7815	191.31	2213	0.9989	15692
8	3456	0.3873	230.94	3456	0.7840	191.44	2192	0.9987	15713
9	3452	0.3891	230.63	3456	0.7843	191.53	2193	0.9989	15802
10	3456	0.3881	230.64	3456	0.7841	191.32	2201	0.9989	15762

Based on these results, it is clear that the third policy is the worst one, and one large case operator can not meet the requirements of all the lines. A Large amount of entities waits for the large casing operation. The daily throughput of the department under the third policy also shows the capacity of one large casing operator. One large casing operator can make maximum of 2200 large cases per day, at average. The first and the second policies are comparable. At this point, both policies give the daily throughput of 3456 large cases approximately. This means that both policies can meet the requirements of the automated part of the packaging line ($4 \times 864 = 3456$). However, the utilization of large casing operator under the first policy is less in comparison to the second policy (i.e., $39\% < 78\%$). Therefore, the second policy is the best in comparison to the other two policies.

As a conclusion, the current policy "2 casing operator and 1 large casing operator per line" is not the best alternative. The policy of "3 casing operator per line and 1 large casing operator per 2 lines" gives the best results in this study. Therefore, the results of this simulation study suggest increasing the number of casing operators from 8 to 12 and decreasing the number of large casing operators from 4 to 2. On the other hand, the daily throughput of the department is increased from $825 \times 4 = 3300$ to 3456 large cases. It must be noted that this study only suggests a new policy based on the technical capability of the operators. However, the applicability of this new policy

depends on whether it is profitable or not. Before applying the new policy, the economic analysis of these changes needs to be made. For example, in this study, the new policy requires 2 additional operators in the packaging department. However, the daily throughput of the department is increased by 156 large cases. To be able to apply this new policy, the profit made by additional large cases per day should be at least equal to the daily cost of the additional operators. If this equality is provided, the new policy can be applied in the packaging department.

In this study, the model was used for evaluating different worker allocation strategies for a given production target under the setup conditions of the packaging lines. The setup conditions of the packaging line refer to the conveyor speed, the product type, V0 feeding rate, the queue capacity, and the number of operators employed per line. If the video cassette demand fluctuates, these setup conditions can be changed to evaluate the performance of alternative policies in meeting the planned production targets.

CHAPTER FOUR

CONCLUSIONS

4. Conclusions

As it was reported in the thesis, simulation is one of the most widely used operations research technique in the area of industrial engineering. Simulation has been used as an operations research technique for decades however the power of this technique was realized only when the advanced hardware and software became widespread in the computer market. Although, the long lead time to develop a simulation model seems a disadvantage, its power and flexibility in analyzing the large and complex systems overcome this disadvantage. Having highly complex relationships between its subsystems, the video cassette manufacturing plant studied in this thesis was just the right environment to be studied by simulation methodology. This plant consists of three main departments. The simulation models of these three departments have been developed in the simulation language of ARENA V2.2 which provides highly flexible modeling environment.

This thesis particularly deals with the use of simulation as a decision support tool at the tactical decision level of the selected application environment. The use of simulation at the tactical decision level involves evaluating the operating policies of a manufacturing system so that given production targets can be met.

The main objective of this study is to evaluate the performance of alternative work force allocation strategies in meeting the production targets for each department under given capacity constraints. The term “capacity constraints” refers to the setup conditions of the production lines, such as the conveyor speeds, the feeding rates of the lines, and the cycle times of the automated machines. Since the production lines are highly automated in this plant, these setup conditions can be changed easily to

regulate the capacity of the production lines according to the fluctuating market demands. These fluctuations in production targets force the managers to change the setup conditions and to apply different work force allocation strategies to meet the production targets.

In the assembly department, 3 assembly lines have been taken into consideration due to the reason that a large amount of time is necessary to carry out one experiment on the whole model. Therefore, the assembly department has been evaluated as two subsystems each having three assembly lines. In the current case, the numbers of workers in the assembly department are “2 transporters per 6 assembly lines”, “1 technician and 1 feeding operator per an assembly line”.

Based on the number of workers employed per line which were given above, the total number of workers employed for the assembly department is equal to $2+1*6+1*6=14$ workers, currently. The main objective of simulating assembly department is to evaluate the performance of alternative work force allocation strategies to meet a predefined production target under given setup conditions. Three policies have been evaluated in the experiment phase of the study; “1 transporter, 1 technician and 1 feeding operator per an assembly line”, “1 transporter, 1 technician and 1 feeding operator per 2 assembly lines”, and “1 transporter, 1 technician and 1 feeding operator per 3 assembly lines”.

The performance measures which have been used to compare alternative policies are “the daily throughput of the first assembly line”, “the utilizations of the workers”, “the utilizations of the main assembly stations”, and “time-in-system”.

The production target for the first line is 30000 VOs per day. The statistical analysis after five replications showed that the base model (to be validated) produces 30000 VOs per day in the 95% confidence level. Following the experimental studies, it has been concluded that there is no difference in daily throughput of the first line between the first and the second policies in %95 confidence level. However, the third policy could not meet the daily production target under same setup conditions of the

assembly lines. This practical result reduces the number of workers employed in the assembly department from 14 to 8 ($2+1*3+1*3$).

The loading department consists of 4 loading lines each having 20 loading machines and one pancake feeding operator. The operating policies of “one operator per a loading line (current policy)”, “one operator per two lines”, and “one operator per 4 lines” have been evaluated to determine the effects of changing the number of pancake feeding operators employed per line on the daily throughput of a loading line. The results of the experiments showed that the daily throughputs of the loading lines are highly sensitive to the number of workers employed per line. Based on the results, if the current policy is substituted by the second policy, the loading lines can not reach the predefined production target (i.e., 44250 loaded VOs) under the same setup conditions of the lines. Therefore, it is decided that the first (current) policy is the best one in comparison to the other two policies.

The packaging department consists of four packaging lines each having two parts. The first part of the packaging line consists of 8 automated packaging machines whereas the second part consists of 2 manual operations. In the current case, the manual operations are carried out by “2 casing and 1 large casing operator per line”. The analysis showed that this worker allocation strategy can not meet the requirements of the automated part of the packaging line. Therefore, the policy of “3 casing and 1 large casing operator per line” has been selected in the first stage of experimentation. In this stage, since it has been observed that the utilization of the large casing operator is very low, the policies of “1 large casing operator per line”, “1 large casing operator per 2 lines”, “1 large casing operator per 4 lines” are evaluated in the second stage of experimentation. Based on the results of the experiments, it has been concluded that one large casing operator is capable of meeting the requirements of 6 casing operators (i.e., 3 casing operators/line * 2 lines). As a conclusion, this simulation study suggested increasing the number of casing operators from 8 to 12 and decreasing the number of large casing operators from 4 to 2.

These flexible simulation models can be used in future to regulate the number of workers employed per line based on fluctuating demand. In this study, the policies have been evaluated for a constant production target and constant setup conditions. Each time that the fluctuating demand requires different setup conditions for changing demand, the production managers can use these simulation models for evaluating different policies to increase or decrease the number of workers employed per line.

The policies that the simulation models suggest should be also analyzed in terms of their profitability. The economic analysis of these policies can be made to decide on whether policies are applicable or not. In addition to the economic analysis, the detailed sensitivity analysis can be made to determine the effects of changing the model variables and parameters on system performance. The sensitivity analysis gives us an idea about the importance of the variables and parameters. The change in more important parameter or variable causes greater effect on system performance and this variable or parameter should be modeled in detail. These models can be used for the purpose of scheduling after some modifications in the model. However, the use of these models for the scheduling purpose requires the simulation models to be initialized based on the current system status of the real system at the beginning of simulation replications. Therefore, technological improvements such as automated data collection machines should be provided by the management. These simulation models can be integrated with the expert system that performs statistical analysis of the input and output data. This kind of an expert system eases the simulation modeling process and reduces the lead time to make a decision.

REFERENCES

- Akhun, M.& Durmuşoğlu, B. (1994). Worker Allocation Problem in Cellular Manufacturing: A Case Study. In Proceedings of the European Simulation Symposium, A.R. Kaylan, A. Lehmann, & T.İ. Ören (Eds.).
- Altıok, T. (1997). Performance Analysis Of Manufacturing Systems. New York: Springer-Verlag.
- Askin, R.G. & Standridge, C.R. (1993). Modeling And Analysis Of Manufacturing Systems. Singapore: John Wiley & Sons.
- Banks, J. (1991). Selecting Simulation Software. In Proceedings of the 1991 Winter Simulation Conference, Barry L. Nelson, W. David Kelton, Gordon M. Clark (Eds.).
- Banks, J., & Gibson, R.R. (1996). Getting Started in Simulation Modeling. IIE Solutions (November), 34-39.
- Banks, J., & Gibson, R.R. (1997). Simulation Modeling. IIE Solutions (February), 26-31.
- Banks, J., & Norman, V.B. (1995). Justifying Simulation in Today's Manufacturing Environment. IIE Solutions (November), 16-19.
- Carrie, A. (1988). Simulation of Manufacturing Systems. New York: John Wiley & Sons.

- Collins, N., & Watson, C.M. (1993). Introduction to Arena™. In Proceedings of the 1993 Winter Simulation Conference, G.W. Evans, M. Mollaghasemi, E.C. Russell, & W.E. Biles (Eds.).
- Corey, P.D. & Clymer, J.R. (1991). Discrete Event Simulation Of Object Movement And Interactions. Simulation, 56:3, 167-174.
- Coskun, R. & Ören, T.İ. (1994). Simulation In Knowledge Based Production Scheduling Systems: A State-Of-The-Art Review. In Proceedings of the European Simulation Symposium, A.R. Kaylan, A. Lehmann, & T.İ. Ören (Eds.).
- Fujimoto, R.M. (1993). Parallel And Distributed Discrete Event Simulation: Algorithms And Applications. In Proceedings of the 1993 Winter Simulation Conference, G.W. Evans, M. Mollaghasemi, E.C. Russell, & W.E. Biles (Eds.).
- Groover, M.P. (1992). Automation, Production Systems, and Computer-Integrated Manufacturing. Englewood Cliffs, New Jersey: Prentice-Hall.
- Haddock, J. (1988). A Simulation Generator For Flexible Manufacturing Systems Design And Control. IIE Transactions, 20:1, 22-31.
- Haider, S.W. & Banks, J. (1986). Simulation Software Products For Analyzing Manufacturing Systems. Industrial Engineering (July), 25-31.
- Houshyar, A. & Nuila, V. (1993). Required Steps For Successful Design And Implementation Of Simulation. Computers And Industrial Engineering, 25, 53-56.
- James, A. & Stoner, F. (1982). Management (2nd ed.). Englewood Cliffs, New Jersey: Prentice-Hall.

- Kiran, A.S., Kaplan, C., & Unal, A.T. (1993). Simulation of Electronics Manufacturing and Assembly Operations: A Survey. In Proceedings of the 1993 Winter Simulation Conference, G.W. Evans, M. Mollaghasemi, E.C. Russell, & W.E. Biles (Eds.).
- Kleijnen, J.P.C. (1995). Theory And Methodology-Verification And Validation Of Simulation Models. European Journal Of Operational Research, 82, 145-162.
- Kusiak, A. & Chen, M. (1988). Expert Systems For Planning And Scheduling Manufacturing Systems. European Journal Of Operational Research, 34, 113-127.
- Law, A.M., & Kelton, W.D. (1991). Simulation Modeling & Analysis. Singapore: McGraw-Hill.
- Law, A.M., & McComas, M.G. (1992). How to select Simulation Software For Manufacturing Applications. Industrial Engineering (July), 29-34.
- Law, A.M., & McComas, M.G. (1991). Secrets Of Successful Simulation Studies. In Proceedings of the 1991 Winter Simulation Conference, Barry L. Nelson, W. David Kelton, Gordon M. Clark (Eds.).
- Lefrancois, P. & Montreuil, B. (1994). An Object-Oriented Knowledge Representation For Intelligent Control Of Manufacturing Workstations. IIE Transactions, 26:1, 11-26.
- Mazziotti, B.W. (1996). Get More Mileage from Flexible Simulation. IIE Solutions (May), 14-22.
- Mellichamp, J.M. & Park, Y.H. (1989). A Statistical Expert System For Simulation Analysis. Simulation, 52:4, 134-139.
- Moore, F.G. (1982). Management in Organization. New York: John Wiley & Sons.

- Nielsen, N.R. (1991). Advances In Simulation Vol.4. In P.A. Fishwick & R. Modjeski (Eds.), Application Of Artificial Intelligence Techniques To Simulation.
- Norman, V.B. (1992). Future Directions in Manufacturing Simulation. Industrial Engineering (July), 36-37.
- O'Keefe R. (1986). Simulation And Expert Systems-A Taxonomy And Some Examples. Simulation, 46, 10-15.
- Ouzrout, Y. & Senoune, R. (1994). Using Knowledge Based System To Object Oriented Manufacturing Systems Simulation. In Proceedings of the European Simulation Symposium, A.R. Kaylan, A. Lehmann, & T.İ. Ören (Eds.).
- Pegden, C. D., Shannon, R.E, & Sadowski, R.P. (1990). Introduction to Simulation Using Siman.
- Porcaro, D. (1996). Simulation Modeling and DOE. IIIE Solutions (September), 24-30.
- Pritsker, A.A.B. (1992). Simulation: The Premier Technique of Industrial Engineering. Industrial Engineering (July), 25-26.
- Rohrer, M. (1996). Visualization And Its Importance In Manufacturing Simulation. Industrial Management (May/June), 15-17
- Salmi M., Harju J. & Porras J. (1994). Computing The Parallel Potential Of Event Oriented Simulation Applications. In Proceedings of the European Simulation Symposium, A.R. Kaylan, A. Lehmann, & T.İ. Ören (Eds.).
- Shannon, R.E. (1992). Introduction to Simulation. In Proceedings of the 1992 Winter Simulation Conference, J.J. Swain, D. Goldsman, R.C. Crain, & J.R. Wilson (Eds.).

Taha, H.A. (1988). Simulation Modeling and Simnet. Englewood Cliffs, New Jersey: Prentice-Hall, Inc. New York: McGraw-Hill.

Thompson, M.B. (1994). Expanding Simulation Beyond Planning and Design. Industrial Engineering (October), 64-67.

Tunali, S. (1997). Evaluation Of Alternative Routing Policies In Scheduling A Job-Shop Type FMS. Computers And Industrial Engineering, 30, 243-250.



APPENDICES

Appendix A.1

7\$	STATION,	L1s1load;
159\$	ASSIGN:	abc=1;
		L1i=1;
10\$	SEIZE,	1:
		L1s1load_r,1;
11\$	EXIT:	L1c2,1;
19\$	DELAY:	L1s1processtime+1/vc(L1s1c);
13\$	ACCESS:	L1s1c,1;
16\$	RELEASE:	L1s1load_r,1;
18\$	CONVEY:	L1s1c,member(L1s1,L1i);
8\$	STATION,	L1s4load;
9\$	SEIZE,	1:
		L1s4load_r,1;
12\$	EXIT:	L1c3,1;
20\$	DELAY:	L1s4processtime+1/vc(L1s4c);
14\$	ACCESS:	L1s4c,1;
15\$	RELEASE:	L1s4load_r,1;
17\$	CONVEY:	L1s4c,member(L1s4,L1i);
21\$	STATION,	L1s1;
22\$	SEIZE,	1:
		L1s1r(L1i),1;
78\$	BRANCH,	2:
		If,L1s1inv(L1i).gt.0,23\$,Yes;
		If,L1s1inv(L1i).eq.L1s1warning(L1i),call,Yes;
		If,L1s1inv(L1i).eq.0,103\$,Yes;
23\$	DELAY:	L1s1processtime;
75\$	ASSIGN:	L1s1inv(L1i)=L1s1inv(L1i)-L1s1u(L1i);
24\$	RELEASE:	L1s1r(L1i),1;
139\$	BRANCH,	1:
		If,L1i.le.4,140\$,Yes;
		Else,141\$,Yes;
140\$	ASSIGN:	L1i=L1i+1;
25\$	CONVEY:	L1s1c,member(L1s1,L1i);
141\$	CONVEY:	L1s1c,L1s1bad;
call	QUEUE,	tr1_Q;
157\$	REQUEST,	1:tr1,,warehouse;
174\$	MOVE:	tr1,member(stock,abc);
203\$	FREE:	tr1;
199\$	QUEUE,	workerq;
200\$	REQUEST,	1:worker,,member(stock,abc):NEXT(204\$);

```

;
;
;
;
;
Model statements for module: Choose 2
204$    TRACE,      -1, "-Choosing from 4 options\n";
222$    BRANCH,      1:
                If,abc==1,76$,Yes:
                If,abc==2,208$,Yes:
                If,abc==3,205$,Yes:
                If,abc==4,211$,Yes;
76$      DELAY:      L1s1unload(L1i);
77$      ASSIGN:     L1s1inv(L1i)=L1s1inv(L1i)+L1s1bat(L1i);
101$     SIGNAL:     L1signal1(L1i),1;
201$     FREE:       worker;
79$      DISPOSE;

208$     DELAY:      L1s2unload(L1n);
209$     ASSIGN:     L1s2inv(L1n)=L1s2inv(L1n)+L1s2bat(L1n);
210$     SIGNAL:     L1signal2(L1n),1:NEXT(201$);

205$     DELAY:      L1s3unload(L1k);
206$     ASSIGN:     L1s3inv(L1k)=L1s3inv(L1k)+L1s3bat(L1k);
207$     SIGNAL:     L1signal3(L1k),1:NEXT(201$);

211$     DELAY:      L1s4unload(L1l);
212$     ASSIGN:     L1s4inv(L1l)=L1s4inv(L1l)+L1s4bat(L1l);
213$     SIGNAL:     L1signal4(L1l),1:NEXT(201$);

;
;
;
;
;
Model statements for module: Duplicate 2
105$     TRACE,      -1, "-Duplicating entities\n";
223$     DUPLICATE:  1,call:NEXT(102$);

102$     WAIT:       L1signal1(L1i),1:NEXT(23$);

26$      STATION,    L1s1end;
47$      SEIZE,      1:
                L1s1end_r,1;
48$      EXIT:       L1s1c,1;
49$      ACCESS:     L1c6,1;
50$      RELEASE:    L1s1end_r,1;
51$      CONVEY:     L1c6,L1s2load;

27$      STATION,    L1s1bad;
28$      SEIZE,      1:
                L1s1bad_r,1;
29$      DELAY:      L1s1processtime;
30$      RELEASE:    L1s1bad_r,1;
32$      BRANCH,     1,1:
                With,.955,31$,Yes:
                With,.045,33$,Yes;
31$      CONVEY:     L1s1c,L1s1end;

33$      EXIT:       L1s1c,1;
34$      DISPOSE;

```

35\$ L1q2	STATION, QUEUE,	L1s6load2; L1s6load2_q:DETACH;
37\$ 38\$	STATION, SEIZE,	L1s4bad; 1: L1s4bad_r,1;
39\$ 40\$ 42\$	DELAY: RELEASE: BRANCH,	L1s4processtime; L1s4bad_r,1; 1,4: With,.955,41\$,Yes; With,.045,43\$,Yes;
41\$	CONVEY:	L1s4c,L1s4end;
43\$ 44\$	EXIT: DISPOSE;	L1s4c,1;
52\$ 53\$	STATION, SEIZE,	L1s2; 1: L1s2r(L1n),1;
91\$	BRANCH,	2: If,L1s2inv(L1n).gt.0,87\$,Yes; If,L1s2inv(L1n).eq.L1s2warning(L1n),call,Yes; If,L1s2inv(L1n).eq.0,103\$,Yes;
87\$ 90\$ 88\$ 142\$	DELAY: ASSIGN: RELEASE: BRANCH,	L1s2processtime; L1s2inv(L1n)=L1s2inv(L1n)-L1s2u(L1n); L1s2r(L1n),1; 1: If,L1n.le.3,143\$,Yes; Else,144\$,Yes;
143\$ 89\$	ASSIGN: CONVEY:	L1n=L1n+1; L1s2c,member(L1s2,L1n);
144\$	CONVEY:	L1s2c,L1s2bad;
;		
;	Model statements for module: Duplicate 4	
;		
103\$ 224\$	TRACE, DUPLICATE:	-1,"-Duplicating entities\n"; 1,call:NEXT(92\$);
92\$	WAIT:	L1signal2(L1n),1:NEXT(87\$);
54\$ L1q1	STATION, QUEUE,	L1s6load1; L1s6load1_q:DETACH;
55\$ 56\$	STATION, SEIZE,	L1s2bad; 1: L1s2bad_r,1;
57\$ 58\$ 60\$	DELAY: RELEASE: BRANCH,	L1s2processtime; L1s2bad_r,1; 1,2: With,.955,59\$,Yes; With,.045,61\$,Yes;
59\$	CONVEY:	L1s2c,L1s2end;

61\$	EXIT:	L1s2c,1;
62\$	DISPOSE;	
63\$	STATION,	L1s2end;
64\$	SEIZE,	1: L1s2end_r,1;
65\$	EXIT:	L1s2c,1;
66\$	ACCESS:	L1c10,1;
67\$	RELEASE:	L1s2end_r,1;
68\$	CONVEY:	L1c10,L1s6load1;
69\$	STATION,	L1s4end;
70\$	SEIZE,	1: L1s4end_r,1;
71\$	EXIT:	L1s4c,1;
72\$	ACCESS:	L1c11,1;
73\$	RELEASE:	L1s4end_r,1;
74\$	CONVEY:	L1c11,L1s6load2;
80\$	STATION,	L1s2load;
158\$	ASSIGN:	L1n=1; abc=2;
81\$	SEIZE,	1: L1s2load_r,1;
82\$	EXIT:	L1c6,1;
86\$	DELAY:	L1s2processtime+1/vc(L1s2c);
83\$	ACCESS:	L1s2c,1;
84\$	RELEASE:	L1s2load_r,1;
85\$	CONVEY:	L1s2c,member(L1s2,L1n);
93\$	STATION,	L1s4;
94\$	SEIZE,	1: L1s4r(L1l),1;
99\$	BRANCH,	2: If,L1s4inv(L1l).gt.0,95\$,Yes; If,L1s4inv(L1l).eq.L1s4warning(L1l),call,Yes; If,L1s4inv(L1l).eq.0,104\$,Yes;
95\$	DELAY:	L1s4processtime;
98\$	ASSIGN:	L1s4inv(L1l)=L1s4inv(L1l)-L1s4u(L1l);
96\$	RELEASE:	L1s4r(L1l),1;
148\$	BRANCH,	1: If,L1l.le.2,149\$,Yes; Else,150\$,Yes;
149\$	ASSIGN:	L1l=L1l+1;
97\$	CONVEY:	L1s4c,member(L1s4,L1l);
150\$	CONVEY:	L1s4c,L1s4bad;
;		
;	Model statements for module:	Duplicate 3
;		
104\$	TRACE,	-1,"-Duplicating entities\n";

225\$	DUPLICATE:	1,call:NEXT(100\$);
100\$	WAIT:	L1signal4(L11):NEXT(95\$);
	MATCH:	L1q1,L1exit1: L1q2,L1exit2: L1q3,L1exit3;
L1exit1 46\$ 161\$	ACCESS: EXIT: CONVEY:	L1c13,1; L1c10,1; L1c13,L1s7;
L1exit2 36\$ 162\$	ACCESS: EXIT: CONVEY:	L1c13,1; L1c11,1; L1c13,L1s7;
L1exit3 131\$ 163\$	ACCESS: EXIT: CONVEY:	L1c13,1; L1c8,1; L1c13,L1s7;
106\$ 160\$ 217\$ 45\$	STATION, COUNT: TALLY: DISPOSE;	L1end; L1end_c,1; time,interval(timein),1;
107\$ L1q3	STATION, QUEUE,	L1s6load3; L1s6load3_q:DETACH;
108\$ 109\$	STATION, SEIZE,	L1s3bad; 1: L1s3bad_r,1;
110\$ 111\$ 113\$	DELAY: RELEASE: BRANCH,	L1s3processtime; L1s3bad_r,1; 1,3: With,.955,112\$,Yes: With,.045,114\$,Yes;
112\$	CONVEY:	L1s3c,L1s3end;
114\$ 115\$	EXIT: DISPOSE;	L1s3c,1;
116\$ 117\$	STATION, SEIZE,	L1s3end; 1: L1s3end_r,1;
118\$ 119\$ 120\$ 121\$	EXIT: ACCESS: RELEASE: CONVEY:	L1s3c,1; L1c8,1; L1s3end_r,1; L1c8,L1s6load3;
122\$ 123\$	STATION, SEIZE,	L1s3; 1: L1s3r(L1k),1;
128\$	BRANCH,	2: If,L1s3inv(L1k).gt.0,124\$,Yes: If,L1s3inv(L1k).eq.L1s3warning(L1k),call,Yes:

```

124$      DELAY:      If,L1s3inv(L1k).eq.0,130$,Yes;
127$      ASSIGN:     L1s3processtime;
125$      RELEASE:    L1s3inv(L1k)=L1s3inv(L1k)-L1s3u(L1k);
145$      BRANCH,     L1s3r(L1k),1;
1:
If,L1k.le.3,146$,Yes;
Else,147$,Yes;
146$      ASSIGN:     L1k=L1k+1;
126$      CONVEY:     L1s3c,member(L1s3,L1k);

147$      CONVEY:     L1s3c,L1s3bad;

;
;
;      Model statements for module: Duplicate 1
;
130$      TRACE,      -1,"-Duplicating entities\n";
226$      DUPLICATE:  1,call:NEXT(129$);

129$      WAIT:       L1signal3(L1k),1:NEXT(124$);


133$      STATION,    L1s3load;
134$      SEIZE,       1;
L1s3load_r,1;
156$      EXIT:       L1c12;
138$      DELAY:       L1s3processtime+1/vc(L1s3c);
135$      ACCESS:      L1s3c,1;
136$      RELEASE:     L1s3load_r,1;
137$      CONVEY:      L1s3c,member(L1s3,L1k);


152$      CREATE,     L1s3batchsize:2.0:MARK(timein);
132$      ASSIGN:      m=L1enters2:
picture=dup:
abc=3:
L1k=1;
153$      QUEUE,       L1c12_q;
154$      ACCESS:      L1c12,1;
155$      CONVEY:      L1c12,L1s3load;


164$      STATION,    L1s7;
165$      EXIT:        L1c13;
166$      COMBINE:     3,First;
167$      ROUTE:       1.88,L1lend;


168$      CREATE,      1:1,1;
169$      SCAN:         nq(L1c2_q)>=10;
170$      ASSIGN:      L1s1batchsizea=0;
171$      SCAN:         nq(L1c2_q)<1;
172$      ASSIGN:      L1s1batchsizea=1:NEXT(169$);


173$      CREATE,      L1s1batchsizea:2:MARK(timein);
0$      ASSIGN:      m=L1enters1:
picture=dup;

```

5\$	QUEUE,	L1c2_q;
1\$	ACCESS:	L1c2,1;
175\$	SIGNAL:	1000,1;
3\$	CONVEY:	L1c2,L1s1load;
178\$	CREATE,	L1s1batchsizeb:2:MARK(timein);
177\$	ASSIGN:	m=L1enters1;
		picture=dup;
6\$	QUEUE,	L1c3_q;
2\$	ACCESS:	L1c3,1;
176\$	WAIT:	1000,1;
151\$	ASSIGN:	abc=4;
		L1l=1;
4\$	CONVEY:	L1c3,L1s4load;
179\$	CREATE,	1:1,1;
180\$	SCAN:	nq(L1c3_q)>=10;
181\$	ASSIGN:	L1s1batchsizeb=0;
182\$	SCAN:	nq(L1c3_q)<1;
183\$	ASSIGN:	L1s1batchsizeb=1:NEXT(180\$);
184\$	CREATE,	1:1,1;
185\$	SCAN:	nq(L1c12_q)>=10;
186\$	ASSIGN:	L1s3batchsize=0;
187\$	SCAN:	nq(L1c12_q)<1;
188\$	ASSIGN:	L1s3batchsize=1:NEXT(185\$);
202\$	CREATE,	1:expo(1200,1);
214\$	ASSIGN:	abc=disc(0.25,1,0.50,2,0.75,3,1,4);
215\$	BRANCH,	1:
		If,(abc==1).and.(mr(L1s1bad_r)==1),189\$,Yes:
		If,(abc==2).and.(mr(L1s2bad_r)==1),190\$,Yes:
		If,(abc==3).and.(mr(L1s3bad_r)==1),191\$,Yes:
		If,(abc==4).and.(mr(L1s4bad_r)==1),192\$,Yes:
		Else,216\$,Yes;
189\$	ASSIGN:	mr(L1s1bad_r)=0;
193\$	QUEUE,	technicianq;
194\$	REQUEST,	1:technician,,member(stock,abc);
195\$	DELAY:	expo(300,3):NEXT(220\$);
:		
:		
:	Model statements for module: Choose 3	
:		
220\$	TRACE,	-1,"-Choosing from 4 options\n";
227\$	BRANCH,	1:
		If,abc==1,198\$,Yes:
		If,abc==2,218\$,Yes:
		If,abc==3,219\$,Yes:
		If,abc==4,221\$,Yes;
198\$	ASSIGN:	mr(L1s1bad_r)=1;
196\$	FREE:	technician;
197\$	DISPOSE;	

218\$	ASSIGN:	mr(L1s2bad_r)=1:NEXT(196\$);
219\$	ASSIGN:	mr(L1s3bad_r)=1:NEXT(196\$);
221\$	ASSIGN:	mr(L1s4bad_r)=1:NEXT(196\$);
190\$	ASSIGN:	mr(L1s2bad_r)=0:NEXT(193\$);
191\$	ASSIGN:	mr(L1s3bad_r)=0:NEXT(193\$);
192\$	ASSIGN:	mr(L1s4bad_r)=0:NEXT(193\$);
216\$	DISPOSE;	



Appendix A.2

PROJECT, ;

ATTRIBUTES: L1i:
 L1n:
 L1k:
 L1l:
 L1signal1(5),50,51,52,53,54:
 L1signal2(4),55,56,57,58:
 L1signal3(4),59,60,61,62:
 L1signal4(3),63,64,65:
 timein, :
 abc,0;

VARIABLES: L1s1processtime,2.15:
 L1s2processtime,2.15:
 L1s3processtime,2:
 L1s4processtime,1.95:
 L1s1batchsizea,1:
 L1s1batchsizeb,1:
 L1s3batchsize,1:
 L1s1inv(5),10000,3500,5000,1666,2000:
 L1s2inv(4),1825,2000,5000,2500:
 L1s3inv(4),4500,2720,2222,2500:
 L1s4inv(3),1920,2000,1280:
 L1s1bat(5),10000,3500,5000,1666,2000:
 L1s2bat(4),1825,2000,5000,2500:
 L1s3bat(4),4500,2720,2222,2500:
 L1s4bat(3),1920,2000,1280:
 L1s1warning(5),1000,350,500,166,200:
 L1s2warning(4),182,200,500,250:
 L1s3warning(4),450,272,222,250:
 L1s4warning(3),192,200,128:
 L1s1u(5),1,1,1,1,1:
 L1s2u(4),1,2,2,2:
 L1s3u(4),2,2,1,2:
 L1s4u(3),2,1,1;

QUEUES: tr1_q,FIFO:
 workerq,FIFO:
 technicianq,FIFO:
 L1c2_q,FIFO:
 L1c3_q,FIFO:
 L1c12_q,FIFO:
 L1s6load1_q,FIFO:
 L1s6load2_q,FIFO:
 L1s6load3_q,FIFO:

PICTURES: default:
 dup;

RESOURCES: L1s1end_r,Capacity(1):
 L1s2end_r,Capacity(1):
 L1s3end_r,Capacity(1):
 L1s4end_r,Capacity(1):
 L1s1r1_r,Capacity(1):
 L1s1r2_r,Capacity(1):

```

L1s1r3_r,Capacity(1,);
L1s1r4_r,Capacity(1,);
L1s1r5_r,Capacity(1,);
L1s2r1_r,Capacity(1,);
L1s2r2_r,Capacity(1,);
L1s2r3_r,Capacity(1,);
L1s2r4_r,Capacity(1,);
L1s3r1_r,Capacity(1,);
L1s3r2_r,Capacity(1,);
L1s3r3_r,Capacity(1,);
L1s3r4_r,Capacity(1,);
L1s4r1_r,Capacity(1,);
L1s4r2_r,Capacity(1,);
L1s4r3_r,Capacity(1,);
L1s1load_r,Capacity(1,);
L1s2load_r,Capacity(1,);
L1s3load_r,Capacity(1,);
L1s4load_r,Capacity(1,);
L1s1bad_r,Capacity(1,);
L1s2bad_r,Capacity(1,);
L1s3bad_r,Capacity(1,);
L1s4bad_r,Capacity(1,);

```

```

STATIONS: L1enters1:
           L1enters2:
           L1end:
           L1s1bad:
           L1s2bad:
           L1s3bad:
           L1s4bad:
           L1s1s:
           L1s2s:
           L1s3s:
           L1s4s:
           L1s1r1:
           L1s1r2:
           L1s1r3:
           L1s1r4:
           L1s1r5:
           L1s2r1:
           L1s2r2:
           L1s2r3:
           L1s2r4:
           L1s3r1:
           L1s3r2:
           L1s3r3:
           L1s3r4:
           L1s4r1:
           L1s4r2:
           L1s4r3:
           L1s1load:
           L1s2load:
           L1s3load:
           L1s4load:
           L1s1end:
           L1s2end:
           L1s3end:

```

```

L1s4end:
L1s6load1:
L1s6load2:
L1s6load3:
L1s7:
warehouse;

DISTANCES:      tech_dst,L1s1s-L1s2s-40,L1s1s-L1s3s-80,L1s1s-L1s4s-120,
                  L1s2s-L1s3s-40, L1s2s-L1s4s-80,L1s3s-L1s4s-40:
tr1_Dst,warehouse-L1s1s-80,warehouse-L1s2s-120,warehouse-L1s3s-160,
warehouse-L1s4s-200;

TRANSPORTERS:   tr1,1,DISTANCE(tr1_Dst),10--,STATION(warehouse)-Active:
                  technician,1,Distance(tech_dst),10.0--,Station(L1s1s)-Active:
                  worker,1,Distance(tech_dst),10.0--,Station(L1s1s)-Active;

SEGMENTS:       L1c2_seg,L1enters1,L1s1load-26:
                  L1c3_seg,L1enters1,L1s4load-122:
                  L1c6_seg,L1s1end,L1s2load-56:
                  L1c8_seg,L1s3end,L1s6load3-75:
                  L1c10_seg,L1s2end,L1s6load1-136:
                  L1c11_seg,L1s4end,L1s6load2-75:
                  L1c12_seg,L1enters2,L1s3load-20:
                  L1c13_seg,L1s6load1,L1s6load2-1,L1s6load3-1,L1s7-1:
                  L1s1c_seg,L1s1load,L1s1r1-1,L1s1r2-1,L1s1r3-1,L1s1r4-1,L1s1r5-1,
                  L1s1bad-1, L1s1end-1,L1s1load-1:
                  L1s2c_seg,L1s2load,L1s2r1-1,L1s2r2-1,L1s2r3-1,L1s2r4-1,L1s2bad-1,
                  L1s2end-1,L1s2load-1:
                  L1s3c_seg,L1s3load,L1s3r1-1,L1s3r2-1,L1s3r3-1,L1s3r4-1,L1s3bad-1,
                  L1s3end-1,L1s3load-1:
                  L1s4c_seg,L1s4load,L1s4r1-1,L1s4r2-1,L1s4r3-1,L1s4bad-1,L1s4end-1,
                  L1s4load-1;

CONVEYORS:       L1c2,L1c2_seg,5,1,Active,1,Accumulating:
                  L1c3,L1c3_seg,5,1,Active,1,Accumulating:
                  L1c6,L1c6_seg,5,0,1,Active,1,Accumulating:
                  L1c8,L1c8_seg,5,0,1,Active,1,Accumulating:
                  L1c10,L1c10_seg,5,0,1,Active,1,Accumulating:
                  L1c11,L1c11_seg,5,0,1,Active,1,Accumulating:
                  L1c12,L1c12_seg,5,0,1,Active,1,Accumulating:
                  L1c13,L1c13_seg,10,1,Active,1,Accumulating:
                  L1s1c,L1s1c_seg,10,1,Active,1,Nonaccumulating:
                  L1s2c,L1s2c_seg,10,1,Active,1,Nonaccumulating:
                  L1s3c,L1s3c_seg,10,1,Active,1,Nonaccumulating:
                  L1s4c,L1s4c_seg,10,1,Active,1,Nonaccumulating:

COUNTERS:       L1end_c,,Replicate;

TALLIES:          time;

DSTATS:          nq(tr1_q):
                  nq(technicianq):
                  nq(workerq):
                  nt(tr1):
                  nt(worker):
                  nt(technician):
                  nr(L1s1bad_r),L1s1 utilization:
                  nr(L1s2bad_r),L1s2 utilization:

```

nr(L1s3bad_r),L1s3 utilization:
nr(L1s4bad_r),L1s4 utilization:

REPLICATE, 5,0.0,88400,Yes,Yes,2000;

EXPRESSIONS:

L1s1unload(5),norm(16.5,1.89),norm(13.3,1.36),norm(9.52,1.01),norm(19,1.86),
norm(14.2,1.53):
L1s2unload(4),norm(20.9,2.21),norm(17,1.66),norm(49,6.33),norm(23.2,2.39):
L1s3unload(4),norm(12.5,1.78),norm(36.1,7.59),norm(44.8,7.77),norm(23.8,4.91):
L1s4unload(3),norm(97.7,16.9),norm(25.3,5.66),norm(85.3,11.4);

SETS:

stock,L1s1s,L1s2s,L1s3s,L1s4s:
L1s1r,L1s1r1_r,L1s1r2_r,L1s1r3_r,L1s1r4_r,L1s1r5_r:
L1s2r,L1s2r1_r,L1s2r2_r,L1s2r3_r,L1s2r4_r:
L1s3r,L1s3r1_r,L1s3r2_r,L1s3r3_r,L1s3r4_r:
L1s4r,L1s4r1_r,L1s4r2_r,L1s4r3_r:
L1s1,L1s1r1,L1s1r2,L1s1r3,L1s1r4,L1s1r5:
L1s2,L1s2r1,L1s2r2,L1s2r3,L1s2r4:
L1s3,L1s3r1,L1s3r2,L1s3r3,L1s3r4:
L1s4,L1s4r1,L1s4r2,L1s4r3;

Appendix B.1

0\$	CREATE,	L3batch:1.5:MARK(timein3);
1\$	ASSIGN:	m=L3enter:
		picture=default:
		l=1:
		lineno=3;
15\$	QUEUE,	L3con1q;
3\$	ACCESS:	L3con1,1;
2\$	CONVEY:	L3con1,member(L3s,l);
4\$	STATION,	L3s;
20\$	BRANCH,	1:
		If,nq(L3q(l)).le.14,9\$,Yes:
		If,nq(L3q(l)).gt.14,25\$,Yes;
9\$	EXIT:	L3con1,1;
5\$	QUEUE,	L3q(l);
17\$	SEIZE,	1:
		L3r(l),1;
18\$	BRANCH,	1:
		If,L3pan(l).gt.0,6\$,Yes:
		If,L3pan(l).le.0,call,Yes;
6\$	DELAY:	37;
16\$	ASSIGN:	L3pan(l)=L3pan(l)-250;
27\$	QUEUE,	L3con2q;
10\$	ACCESS:	L3con2,1;
7\$	RELEASE:	L3r(l),1;
11\$	CONVEY:	L3con2,L3end;
call	QUEUE,	tr1q;
79\$	REQUEST,	1:tr1,,L1pancake:NEXT(124\$);
;	Model statements for module: Choose 1	
;		
124\$	TRACE,	-1,"-Choosing from 4 options\n";
134\$	BRANCH,	1:
		If,lineno==1,123\$,Yes:
		If,lineno==2,125\$,Yes:
		If,lineno==3,130\$,Yes:
		If,lineno==4,132\$,Yes;
123\$	MOVE:	tr1,member(L1s,i);
80\$	DELAY:	normal(54.8,5.11,1):NEXT(L1);
L1	ASSIGN:	L1pan(i)=12000;
81\$	FREE:	tr1;
66\$	DELAY:	37;
76\$	ASSIGN:	L1pan(i)=L1pan(i)-250;
89\$	QUEUE,	L1con2q;
70\$	ACCESS:	L1con2,1;
67\$	RELEASE:	L1r(i),1;
71\$	CONVEY:	L1con2,L1end;
125\$	MOVE:	tr1,member(L2s,k);
126\$	DELAY:	normal(54.8,5.11,2):NEXT(L2);

L2	ASSIGN:	L2pan(k)=12000;
111\$	FREE:	tr1;
98\$	DELAY:	37;
108\$	ASSIGN:	L2pan(k)=L2pan(k)-250;
119\$	QUEUE,	L2con2q;
102\$	ACCESS:	L2con2,1;
99\$	RELEASE:	L2r(k),1;
103\$	CONVEY:	L2con2,L2end;
130\$	MOVE:	tr1,member(L3s,l);
131\$	DELAY:	normal(54.8,5.11,3):NEXT(L3);
L3	ASSIGN:	L3pan(l)=12000;
19\$	FREE:	tr1:NEXT(6\$);
132\$	MOVE:	tr1,member(L4s,t);
133\$	DELAY:	normal(54.8,5.11,4):NEXT(L4);
L4	ASSIGN:	L4pan(t)=12000;
49\$	FREE:	tr1;
36\$	DELAY:	37;
46\$	ASSIGN:	L4pan(t)=L4pan(t)-250;
57\$	QUEUE,	L4con2q;
40\$	ACCESS:	L4con2,1;
37\$	RELEASE:	L4r(t),1;
41\$	CONVEY:	L4con2,L4end;
25\$	BRANCH,	1:
		If,l.lt.20,22\$,Yes:
		If,l.eq.20,28\$,Yes;
22\$	ASSIGN:	l=l+1;
21\$	CONVEY:	L3con1,member(L3s,l);
28\$	ASSIGN:	L3batch=0;
24\$	STOP:	L3con1;
26\$	SCAN:	nq(L3q(20)).eq.10;
29\$	ASSIGN:	L3batch=1;
23\$	START:	L3con1:NEXT(9\$);
12\$	STATION,	L3end;
13\$	EXIT:	L3con2,1;
14\$	COUNT:	L3jobs,1;
128\$	TALLY:	time3,interval(timein3),45000;
8\$	DISPOSE;	
30\$	CREATE,	L4batch:1.5:MARK(timein4);
31\$	ASSIGN:	m=L4enter:
		picture=default:
		t=1:
		lineno=4;
45\$	QUEUE,	L4con1q;
33\$	ACCESS:	L4con1,1;
32\$	CONVEY:	L4con1,member(L4s,t);
34\$	STATION,	L4s;

50\$	BRANCH,	1: If,nq(L4q(t)).le.14,39\$,Yes: If,nq(L4q(t)).gt.14,55\$,Yes;
39\$	EXIT:	L4con1,1;
35\$	QUEUE,	L4q(t);
47\$	SEIZE,	1: L4r(t),1;
48\$	BRANCH,	1: If,L4pan(t).gt.0,36\$,Yes: If,L4pan(t).le.0,call,Yes;
55\$	BRANCH,	1: If,t.lt.20,52\$,Yes: If,t.eq.20,58\$,Yes;
52\$	ASSIGN:	t=t+1;
51\$	CONVEY:	L4con1,member(L4s,t);
58\$	ASSIGN:	L4batch=0;
54\$	STOP:	L4con1;
56\$	SCAN:	nq(L4q(20)).eq.10;
59\$	ASSIGN:	L4batch=1;
53\$	START:	L4con1:NEXT(39\$);
42\$	STATION,	L4end;
43\$	EXIT:	L4con2,1;
44\$	COUNT:	L4jobs,1;
129\$	TALLY:	time4,interval(timein4),45000;
38\$	DISPOSE;	
60\$	CREATE,	L1batch:1.5:MARK(timein1);
61\$	ASSIGN:	m=L1enter: picture=default: i=1: lineno=1;
75\$	QUEUE,	L1con1q;
63\$	ACCESS:	L1con1,1;
62\$	CONVEY:	L1con1,member(L1s,i);
64\$	STATION,	L1s;
82\$	BRANCH,	1: If,nq(L1q(i)).le.14,69\$,Yes: If,nq(L1q(i)).gt.14,87\$,Yes;
69\$	EXIT:	L1con1,1;
65\$	QUEUE,	L1q(i);
77\$	SEIZE,	1: L1r(i),1;
78\$	BRANCH,	1: If,L1pan(i).gt.0,66\$,Yes: If,L1pan(i).le.0,call,Yes;
87\$	BRANCH,	1: If,i.lt.20,84\$,Yes: If,i.eq.20,90\$,Yes;
84\$	ASSIGN:	i=i+1;
83\$	CONVEY:	L1con1,member(L1s,i);
90\$	ASSIGN:	L1batch=0;

86\$	STOP:	L1con1;
88\$	SCAN:	nq(L1q(20)).eq.10;
91\$	ASSIGN:	L1batch=1;
85\$	START:	L1con1:NEXT(69\$);
72\$	STATION,	L1end;
73\$	EXIT:	L1con2,1;
74\$	COUNT:	L1jobs,1;
122\$	TALLY:	time1,interval(timein1),45000;
68\$	DISPOSE;	
92\$	CREATE,	L2batch:1.5:MARK(timein2);
93\$	ASSIGN:	m=L2enter;
		picture=default;
		k=1;
		lineno=2;
107\$	QUEUE,	L2con1q;
95\$	ACCESS:	L2con1,1;
94\$	CONVEY:	L2con1,member(L2s,k);
96\$	STATION,	L2s;
112\$	BRANCH,	1:
		If,nq(L2q(k)).le.14,101\$,Yes;
		If,nq(L2q(k)).gt.14,117\$,Yes;
101\$	EXIT:	L2con1,1;
97\$	QUEUE,	L2q(k);
109\$	SEIZE,	1:
		L2r(k),1;
110\$	BRANCH,	1:
		If,L2pan(k).gt.0,98\$,Yes;
		If,L2pan(k).le.0,call,Yes;
117\$	BRANCH,	1:
		If,k.lt.20,114\$,Yes;
		If,k.eq.20,120\$,Yes;
114\$	ASSIGN:	k=k+1;
113\$	CONVEY:	L2con1,member(L2s,k);
120\$	ASSIGN:	L2batch=0;
116\$	STOP:	L2con1;
118\$	SCAN:	nq(L2q(20)).eq.10;
121\$	ASSIGN:	L2batch=1;
115\$	START:	L2con1:NEXT(101\$);
104\$	STATION,	L2end;
105\$	EXIT:	L2con2,1;
106\$	COUNT:	L2jobs,1;
127\$	TALLY:	time2,interval(timein2),45000;
100\$	DISPOSE;	

Appendix B.2

PROJECT, ;

ATTRIBUTES:

timein1:

timein2:

timein3:

timein4:

lineno:

i:

k:

l:

t;

VARIABLES:

L1pan(20), 12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000:

L2pan(20), 12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000:

L3pan(20), 12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000:

L4pan(20), 12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000,
12000,12000,12000,12000,12000:

L1batch,1:

L2batch,1:

L3batch,1:

L4batch,1;

QUEUES:

L1q1,FIFO:	L2q1,FIFO:	L3q1,FIFO:	L4q1,FIFO:	L1con1q,FIFO:
L1q2,FIFO:	L2q2,FIFO:	L3q2,FIFO:	L4q2,FIFO:	L1con2q,FIFO:
L1q3,FIFO:	L2q3,FIFO:	L3q3,FIFO:	L4q3,FIFO:	L2con1q,FIFO:
L1q4,FIFO:	L2q4,FIFO:	L3q4,FIFO:	L4q4,FIFO:	L2con2q,FIFO:
L1q5,FIFO:	L2q5,FIFO:	L3q5,FIFO:	L4q5,FIFO:	L3con1q,FIFO:
L1q6,FIFO:	L2q6,FIFO:	L3q6,FIFO:	L4q6,FIFO:	L3con2q,FIFO:
L1q7,FIFO:	L2q7,FIFO:	L3q7,FIFO:	L4q7,FIFO:	L4con1q,FIFO:
L1q8,FIFO:	L2q8,FIFO:	L3q8,FIFO:	L4q8,FIFO:	L4con2q,FIFO:
L1q9,FIFO:	L2q9,FIFO:	L3q9,FIFO:	L4q9,FIFO:	tr1q,FIFO;
L1q10,FIFO:	L2q10,FIFO:	L3q10,FIFO:	L4q10,FIFO:	
L1q11,FIFO:	L2q11,FIFO:	L3q11,FIFO:	L4q11,FIFO:	
L1q12,FIFO:	L2q12,FIFO:	L3q12,FIFO:	L4q12,FIFO:	
L1q13,FIFO:	L2q13,FIFO:	L3q13,FIFO:	L4q13,FIFO:	
L1q14,FIFO:	L2q14,FIFO:	L3q14,FIFO:	L4q14,FIFO:	
L1q15,FIFO:	L2q15,FIFO:	L3q15,FIFO:	L4q15,FIFO:	
L1q16,FIFO:	L2q16,FIFO:	L3q16,FIFO:	L4q16,FIFO:	
L1q17,FIFO:	L2q17,FIFO:	L3q17,FIFO:	L4q17,FIFO:	
L1q18,FIFO:	L2q18,FIFO:	L3q18,FIFO:	L4q18,FIFO:	
L1q19,FIFO:	L2q19,FIFO:	L3q19,FIFO:	L4q19,FIFO:	

L1q20,FIFO: L2q20,FIFO: L3q20,FIFO: L4q20,FIFO:

PICTURES: default;

RESOURCES:

L1r1,Capacity(1,):	L2r1,Capacity(1,):	L3r1,Capacity(1,):	L4r1,Capacity(1,):
L1r2,Capacity(1,):	L2r2,Capacity(1,):	L3r2,Capacity(1,):	L4r2,Capacity(1,):
L1r3,Capacity(1,):	L2r3,Capacity(1,):	L3r3,Capacity(1,):	L4r3,Capacity(1,):
L1r4,Capacity(1,):	L2r4,Capacity(1,):	L3r4,Capacity(1,):	L4r4,Capacity(1,):
L1r5,Capacity(1,):	L2r5,Capacity(1,):	L3r5,Capacity(1,):	L4r5,Capacity(1,):
L1r6,Capacity(1,):	L2r6,Capacity(1,):	L3r6,Capacity(1,):	L4r6,Capacity(1,):
L1r7,Capacity(1,):	L2r7,Capacity(1,):	L3r7,Capacity(1,):	L4r7,Capacity(1,):
L1r8,Capacity(1,):	L2r8,Capacity(1,):	L3r8,Capacity(1,):	L4r8,Capacity(1,):
L1r9,Capacity(1,):	L2r9,Capacity(1,):	L3r9,Capacity(1,):	L4r9,Capacity(1,):
L1r10,Capacity(1,):	L2r10,Capacity(1,):	L3r10,Capacity(1,):	L4r10,Capacity(1,):
L1r11,Capacity(1,):	L2r11,Capacity(1,):	L3r11,Capacity(1,):	L4r11,Capacity(1,):
L1r12,Capacity(1,):	L2r12,Capacity(1,):	L3r12,Capacity(1,):	L4r12,Capacity(1,):
L1r13,Capacity(1,):	L2r13,Capacity(1,):	L3r13,Capacity(1,):	L4r13,Capacity(1,):
L1r14,Capacity(1,):	L2r14,Capacity(1,):	L3r14,Capacity(1,):	L4r14,Capacity(1,):
L1r15,Capacity(1,):	L2r15,Capacity(1,):	L3r15,Capacity(1,):	L4r15,Capacity(1,):
L1r16,Capacity(1,):	L2r16,Capacity(1,):	L3r16,Capacity(1,):	L4r16,Capacity(1,):
L1r17,Capacity(1,):	L2r17,Capacity(1,):	L3r17,Capacity(1,):	L4r17,Capacity(1,):
L1r18,Capacity(1,):	L2r18,Capacity(1,):	L3r18,Capacity(1,):	L4r18,Capacity(1,):
L1r19,Capacity(1,):	L2r19,Capacity(1,):	L3r19,Capacity(1,):	L4r19,Capacity(1,):
L1r20,Capacity(1,):	L2r20,Capacity(1,):	L3r20,Capacity(1,):	L4r20,Capacity(1,):

STATIONS:

L1s1:	L2s1:	L3s1:	L4s1:	L1enter:
L1s2:	L2s2:	L3s2:	L4s2:	L1end:
L1s3:	L2s3:	L3s3:	L4s3:	L2enter:
L1s4:	L2s4:	L3s4:	L4s4:	L2end:
L1s5:	L2s5:	L3s5:	L4s5:	L3enter:
L1s6:	L2s6:	L3s6:	L4s6:	L3end:
L1s7:	L2s7:	L3s7:	L4s7:	L4enter:
L1s8:	L2s8:	L3s8:	L4s8:	L4end:
L1s9:	L2s9:	L3s9:	L4s9:	L1pancake;
L1s10:	L2s10:	L3s10:	L4s10:	
L1s11:	L2s11:	L3s11:	L4s11:	
L1s12:	L2s12:	L3s12:	L4s12:	
L1s13:	L2s13:	L3s13:	L4s13:	
L1s14:	L2s14:	L3s14:	L4s14:	
L1s15:	L2s15:	L3s15:	L4s15:	
L1s16:	L2s16:	L3s16:	L4s16:	
L1s17:	L2s17:	L3s17:	L4s17:	
L1s18:	L2s18:	L3s18:	L4s18:	
L1s19:	L2s19:	L3s19:	L4s19:	
L1s20:	L2s20:	L3s20:	L4s20:	

DISTANCES: tr1dis,L1pancake-L1s1-5,L1pancake-L1s2-10,L1pancake-L1s3-15,
 L1pancake-L1s4-20,L1pancake-L1s5-25,L1pancake-L1s6-30,
 L1pancake-L1s7-35,L1pancake-L1s8-40,L1pancake-L1s9-45,
 L1pancake-L1s10-50,L1pancake-L1s11-55,L1pancake-L1s12-60,
 L1pancake-L1s13-65,L1pancake-L1s14-70,L1pancake-L1s15-75,
 L1pancake-L1s16-80,L1pancake-L1s17-85,L1pancake-L1s18-90,
 L1pancake-L1s19-95,L1pancake-L1s20-100,
 L1pancake-L2s1-15,L1pancake-L2s2-20,L1pancake-L2s3-25,
 L1pancake-L2s4-30,L1pancake-L2s5-35,L1pancake-L2s6-40,
 L1pancake-L2s7-45,L1pancake-L2s8-50,L1pancake-L2s9-55,

L1pancake-L2s10-60,L1pancake-L2s11-65,L1pancake-L2s12-70,
 L1pancake-L2s13-75,L1pancake-L2s14-80,L1pancake-L2s15-85,
 L1pancake-L2s16-90,L1pancake-L2s17-95,L1pancake-L2s18-100,
 L1pancake-L2s19-105,L1pancake-L2s20-110,
 L1pancake-L3s1-25,L1pancake-L3s2-30,L1pancake-L3s3-35,
 L1pancake-L3s4-40,L1pancake-L3s5-45,L1pancake-L3s6-50,
 L1pancake-L3s7-55,L1pancake-L3s8-60,L1pancake-L3s9-65,
 L1pancake-L3s10-70,L1pancake-L3s11-75,L1pancake-L3s12-80,
 L1pancake-L3s13-85,L1pancake-L3s14-90,L1pancake-L3s15-95,
 L1pancake-L3s16-100,L1pancake-L3s17-105,L1pancake-L3s18-110,
 L1pancake-L3s19-115,L1pancake-L3s20-120,
 L1pancake-L4s1-35,L1pancake-L4s2-40,L1pancake-L4s3-45,
 L1pancake-L4s4-50,L1pancake-L4s5-55,L1pancake-L4s6-60,
 L1pancake-L4s7-65,L1pancake-L4s8-70,L1pancake-L4s9-75,
 L1pancake-L4s10-80,L1pancake-L4s11-85,L1pancake-L4s12-90,
 L1pancake-L4s13-95,L1pancake-L4s14-100,L1pancake-L4s15-105,
 L1pancake-L4s16-110,L1pancake-L4s17-115,L1pancake-L4s18-120,
 L1pancake-L4s19-125,L1pancake-L4s20-130;

TRANSPORTERS: tr1,1,Distance(tr1dis),10—,STATION(L1pancake)-Active;

SEGMENTS:

L1con1seg,L1enter,L1s1-7,L1s2-7,L1s3-7,L1s4-7,L1s5-7,L1s6-7,
 L1s7-7,L1s8-7,L1s9-7,L1s10-7,L1s11-7,L1s12-7,L1s13-7,
 L1s14-7,L1s15-7,L1s16-7,L1s17-7,L1s18-7,L1s19-7,L1s20-7:
 L1con2seg,L1enter,L1s1-7,L1s2-7,L1s3-7,L1s4-7,L1s5-7,L1s6-7,
 L1s7-7,L1s8-7,L1s9-7,L1s10-7,L1s11-7,L1s12-7,L1s13-7,
 L1s14-7,L1s15-7,L1s16-7,L1s17-7,L1s18-7,L1s19-7,
 L1s20-7,L1end-7:
 L2con1seg,L2enter,L2s1-7,L2s2-7,L2s3-7,L2s4-7,L2s5-7,L2s6-7,
 L2s7-7,L2s8-7,L2s9-7,L2s10-7,L2s11-7,L2s12-7,L2s13-7,
 L2s14-7,L2s15-7,L2s16-7,L2s17-7,L2s18-7,L2s19-7,L2s20-7:
 L2con2seg,L2enter,L2s1-7,L2s2-7,L2s3-7,L2s4-7,L2s5-7,L2s6-7,
 L2s7-7,L2s8-7,L2s9-7,L2s10-7,L2s11-7,L2s12-7,L2s13-7,
 L2s14-7,L2s15-7,L2s16-7,L2s17-7,L2s18-7,L2s19-7,
 L2s20-7,L2end-7:
 L3con1seg,L3enter,L3s1-7,L3s2-7,L3s3-7,L3s4-7,L3s5-7,L3s6-7,
 L3s7-7,L3s8-7,L3s9-7,L3s10-7,L3s11-7,L3s12-7,L3s13-7,
 L3s14-7,L3s15-7,L3s16-7,L3s17-7,L3s18-7,L3s19-7,L3s20-7:
 L3con2seg,L3enter,L3s1-7,L3s2-7,L3s3-7,L3s4-7,L3s5-7,L3s6-7,
 L3s7-7,L3s8-7,L3s9-7,L3s10-7,L3s11-7,L3s12-7,L3s13-7,
 L3s14-7,L3s15-7,L3s16-7,L3s17-7,L3s18-7,L3s19-7,
 L3s20-7,L3end-7:
 L4con1seg,L4enter,L4s1-7,L4s2-7,L4s3-7,L4s4-7,L4s5-7,L4s6-7,
 L4s7-7,L4s8-7,L4s9-7,L4s10-7,L4s11-7,L4s12-7,L4s13-7,
 L4s14-7,L4s15-7,L4s16-7,L4s17-7,L4s18-7,L4s19-7,L4s20-7:
 L4con2seg,L4enter,L4s1-7,L4s2-7,L4s3-7,L4s4-7,L4s5-7,L4s6-7,
 L4s7-7,L4s8-7,L4s9-7,L4s10-7,L4s11-7,L4s12-7,L4s13-7,
 L4s14-7,L4s15-7,L4s16-7,L4s17-7,L4s18-7,L4s19-7,
 L4s20-7,L4end-7;

CONVEYORS:

L1con1,L1con1seg,5,1,Active,1,Accumulating:
 L1con2,L1con2seg,5,1,Active,1,Accumulating:
 L2con1,L2con1seg,5,1,Active,1,Accumulating:
 L2con2,L2con2seg,5,1,Active,1,Accumulating:
 L3con1,L3con1seg,5,1,Active,1,Accumulating:
 L3con2,L3con2seg,5,1,Active,1,Accumulating:
 L4con1,L4con1seg,5,1,Active,1,Accumulating:
 L4con2,L4con2seg,5,1,Active,1,Accumulating:

COUNTERS: L1jobs,,Replicate:
 L2jobs,,Replicate:
 L3jobs,,Replicate:
 L4jobs,,Replicate;

TALLIES: time1:
 time2:
 time3:
 time4;

DSTATS: nq(tr1q):
 nq(L1con1q):
 nq(L1con2q):
 nq(L2con1q):
 nq(L2con2q):
 nq(L3con1q):
 nq(L3con2q):
 nq(L4con1q):
 nq(L4con2q):
 nt(tr1):
 nr(L1r1):
 nr(L1r10):
 nr(L1r20):
 nr(L2r1):
 nr(L2r10):
 nr(L2r20):
 nr(L3r1):
 nr(L3r10):
 nr(L3r20):
 nr(L4r1):
 nr(L4r10):
 nr(L4r20);

REPLICATE, 10,0.0,92400,Yes,Yes,6000;

SETS: L1s,L1s1,L1s2,L1s3,L1s4,L1s5,L1s6,L1s7,L1s8,L1s9,L1s10,
 L1s11,L1s12,L1s13,L1s14,L1s15,L1s16,L1s17,L1s18,L1s19,L1s20:
 L1q,L1q1,L1q2,L1q3,L1q4,L1q5,L1q6,L1q7,L1q8,L1q9,L1q10,
 L1q11,L1q12,L1q13,L1q14,L1q15,L1q16,L1q17,L1q18,L1q19,L1q20:
 L1r,L1r1,L1r2,L1r3,L1r4,L1r5,L1r6,L1r7,L1r8,L1r9,L1r10,
 L1r11,L1r12,L1r13,L1r14,L1r15,L1r16,L1r17,L1r18,L1r19,L1r20:
 L2s,L2s1,L2s2,L2s3,L2s4,L2s5,L2s6,L2s7,L2s8,L2s9,L2s10,
 L2s11,L2s12,L2s13,L2s14,L2s15,L2s16,L2s17,L2s18,L2s19,L2s20:
 L2q,L2q1,L2q2,L2q3,L2q4,L2q5,L2q6,L2q7,L2q8,L2q9,L2q10,
 L2q11,L2q12,L2q13,L2q14,L2q15,L2q16,L2q17,L2q18,L2q19,L2q20:
 L2r,L2r1,L2r2,L2r3,L2r4,L2r5,L2r6,L2r7,L2r8,L2r9,L2r10,
 L2r11,L2r12,L2r13,L2r14,L2r15,L2r16,L2r17,L2r18,L2r19,L2r20:
 L3s,L3s1,L3s2,L3s3,L3s4,L3s5,L3s6,L3s7,L3s8,L3s9,L3s10,
 L3s11,L3s12,L3s13,L3s14,L3s15,L3s16,L3s17,L3s18,L3s19,L3s20:
 L3q,L3q1,L3q2,L3q3,L3q4,L3q5,L3q6,L3q7,L3q8,L3q9,L3q10,
 L3q11,L3q12,L3q13,L3q14,L3q15,L3q16,L3q17,L3q18,L3q19,L3q20:
 L3r,L3r1,L3r2,L3r3,L3r4,L3r5,L3r6,L3r7,L3r8,L3r9,L3s10,
 L3r11,L3r12,L3r13,L3r14,L3r15,L3r16,L3r17,L3r18,L3r19,L3r20:
 L4s,L4s1,L4s2,L4s3,L4s4,L4s5,L4s6,L4s7,L4s8,L4s9,L4s10,
 L4s11,L4s12,L4s13,L4s14,L4s15,L4s16,L4s17,L4s18,L4s19,L4s20:
 L4q,L4q1,L4q2,L4q3,L4q4,L4q5,L4q6,L4q7,L4q8,L4q9,L4q10,
 L4q11,L4q12,L4q13,L4q14,L4q15,L4q16,L4q17,L4q18,L4q19,L4q20:

L4r,L4r1,L4r2,L4r3,L4r4,L4r5,L4r6,L4r7,L4r8,L4r9,L4s10,
L4r11,L4r12,L4r13,L4r14,L4r15,L4r16,L4r17,L4r18,L4r19,L4r20:



Appendix C.1

0\$	CREATE,	L1batchsize:1.5:MARK(timein1);
1\$	ASSIGN:	m=L1enter;
		picture=default;
		i=1;
9\$	QUEUE,	L1con1q;
3\$	ACCESS:	L1con1,1;
2\$	CONVEY:	L1con1,member(L1s,i);
4\$	STATION,	L1s;
5\$	QUEUE,	L1q(i);
10\$	SEIZE,	1:
		L1r(i),1;
6\$	DELAY:	L1pan(i);
7\$	RELEASE:	L1r(i),1;
11\$	BRANCH,	1:
		If,i.le.7,13\$,Yes:
		Else,16\$,Yes;
13\$	ASSIGN:	i=i+1;
12\$	CONVEY:	L1con1,member(L1s,i);
16\$	CONVEY:	L1con1,L1end;
14\$	STATION,	L1end;
120\$	BRANCH,	1:
		If,nq(L1m1q)<=10,17\$,Yes:
		Else,121\$,Yes;
17\$	EXIT:	L1con1;
80\$	COMBINE:	10,First;
81\$	PICKQ,	cyc:
		L1m1:
		L1m2:
		L1m3;
L1m1	QUEUE,	L1m1q;
82\$	SEIZE,	1:
		L1m1r,1;
84\$	DELAY:	normal(41.9,4.23,1);
86\$	RELEASE:	L1m1r,1;
91\$	COMBINE:	5,First;
88\$	QUEUE,	L1m4q;
89\$	SEIZE,	1:
		L1m4r,1;
90\$	DELAY:	normal(39.1,6.06);
92\$	RELEASE:	L1m4r,1;
8\$	COUNT:	L1jobs,1;
119\$	TALLY:	time1,interval(timein1),4000;
15\$	DISPOSE;	
L1m2	QUEUE,	L1m2q;
83\$	SEIZE,	1:
		L1m2r,1;
85\$	DELAY:	normal(41.9,4.23,2);
87\$	RELEASE:	L1m2r,1:NEXT(91\$);
L1m3	QUEUE,	L1m3q;

122\$	SEIZE,	1:
		L1m3r,1;
123\$	DELAY:	normal(41.9,4.23,3);
124\$	RELEASE:	L1m3r,1:NEXT(91\$);
121\$	SCAN:	nq(L1m1q)=0:NEXT(17\$);
18\$	CREATE,	L2batchsize:1.5:MARK(timein1);
19\$	ASSIGN:	m=L2enter:
		picture=default:
		k=1;
26\$	QUEUE,	L2con1q;
21\$	ACCESS:	L2con1,1;
20\$	CONVEY:	L2con1,member(L2s,k);
22\$	STATION,	L2s;
23\$	QUEUE,	L2q(k);
27\$	SEIZE,	1:
		L2r(k),1;
24\$	DELAY:	L2pan(k);
25\$	RELEASE:	L2r(k),1;
28\$	BRANCH,	1:
		If,k.le.7,30\$,Yes:
		Else,31\$,Yes;
30\$	ASSIGN:	k=k+1;
29\$	CONVEY:	L2con1,member(L2s,k);
31\$	CONVEY:	L2con1,L2end;
32\$	CREATE,	L3batchsize:1.5:MARK(timein1);
33\$	ASSIGN:	m=L3enter:
		picture=default:
		n=1;
40\$	QUEUE,	L3con1q;
35\$	ACCESS:	L3con1,1;
34\$	CONVEY:	L3con1,member(L3s,n);
36\$	STATION,	L3s;
37\$	QUEUE,	L3q(n);
41\$	SEIZE,	1:
		L3r(n),1;
38\$	DELAY:	L3pan(n);
39\$	RELEASE:	L3r(n),1;
42\$	BRANCH,	1:
		If,n.le.7,44\$,Yes:
		Else,45\$,Yes;
44\$	ASSIGN:	n=n+1;
43\$	CONVEY:	L3con1,member(L3s,n);
45\$	CONVEY:	L3con1,L3end;
46\$	CREATE,	L4batchsize:1.5:MARK(timein1);
47\$	ASSIGN:	m=L4enter:

		picture=default:
		l=1;
54\$	QUEUE,	L4con1q;
49\$	ACCESS:	L4con1,1;
48\$	CONVEY:	L4con1,member(L4s,l);
50\$	STATION,	L4s;
51\$	QUEUE,	L4q(l);
55\$	SEIZE,	l:
		L4r(l),1;
52\$	DELAY:	L4pan(l);
53\$	RELEASE:	L4r(l),1;
56\$	BRANCH,	l:
		If,l.le.7,58\$,Yes:
		Else,59\$,Yes;
58\$	ASSIGN:	l=l+1;
57\$	CONVEY:	L4con1,member(L4s,l);
59\$	CONVEY:	L4con1,L4end;
60\$	CREATE,	l:1,1;
61\$	SCAN:	nq(L1con1q).eq.10;
63\$	ASSIGN:	L1batchsize=0;
62\$	SCAN:	nq(L1con1q).eq.1;
64\$	ASSIGN:	L1batchsize=1:NEXT(61\$);
65\$	CREATE,	l:1,1;
66\$	SCAN:	nq(L2con1q).eq.10;
68\$	ASSIGN:	L2batchsize=0;
67\$	SCAN:	nq(L2con1q).eq.1;
69\$	ASSIGN:	L2batchsize=1:NEXT(66\$);
70\$	CREATE,	l:1,1;
71\$	SCAN:	nq(L3con1q).eq.10;
73\$	ASSIGN:	L3batchsize=0;
72\$	SCAN:	nq(L3con1q).eq.1;
74\$	ASSIGN:	L3batchsize=1:NEXT(71\$);
75\$	CREATE,	l:1,1;
76\$	SCAN:	nq(L4con1q).eq.10;
78\$	ASSIGN:	L4batchsize=0;
77\$	SCAN:	nq(L4con1q).eq.1;
79\$	ASSIGN:	L4batchsize=1:NEXT(76\$);
93\$	STATION,	L2end;
125\$	BRANCH,	l:
		If,nq(L2m1q)<=10,94\$,Yes:
		Else,126\$,Yes;
94\$	EXIT:	L2con1;
95\$	COMBINE:	10,First;
96\$	PICKQ,	cyc:
		L2m1:

L2m1	QUEUE,	L2m2:
97\$	SEIZE,	L2m3;
		L2m1q;
		1:
99\$	DELAY:	L2m1r,1;
100\$	RELEASE:	normal(41.9,4.23,4);
		L2m1r,1:NEXT(91\$);
L2m2	QUEUE,	L2m2q;
98\$	SEIZE,	1:
		L2m2r,1;
129\$	DELAY:	normal(41.9,4.23,5);
101\$	RELEASE:	L2m2r,1:NEXT(91\$);
L2m3	QUEUE,	L2m3q;
127\$	SEIZE,	1:
		L2m3r,1;
130\$	DELAY:	normal(41.9,4.23,6);
128\$	RELEASE:	L2m3r,1:NEXT(91\$);
126\$	SCAN:	nq(L2m1q)=0:NEXT(94\$);
102\$	STATION,	L3end;
140\$	BRANCH,	1:
		If,nq(L3m1q)<=10,103\$,Yes:
		Else,141\$,Yes;
103\$	EXIT:	L3con1;
104\$	COMBINE:	10,First;
105\$	PICKQ,	cyc:
		L3m1:
		L3m2:
		L3m3;
L3m1	QUEUE,	L3m1q;
106\$	SEIZE,	1:
		L3m1r,1;
108\$	DELAY:	normal(41.9,4.23,7);
109\$	RELEASE:	L3m1r,1:NEXT(91\$);
L3m2	QUEUE,	L3m2q;
107\$	SEIZE,	1:
		L3m2r,1;
133\$	DELAY:	normal(41.9,4.23,7);
110\$	RELEASE:	L3m2r,1:NEXT(91\$);
L3m3	QUEUE,	L3m3q;
131\$	SEIZE,	1:
		L3m3r,1;
134\$	DELAY:	normal(41.9,4.23,7);
132\$	RELEASE:	L3m3r,1:NEXT(91\$);
141\$	SCAN:	nq(L3m1q)=0:NEXT(103\$);
111\$	STATION,	L4end;
142\$	BRANCH,	1:
		If,nq(L4m1q)<=10,112\$,Yes:
		Else,143\$,Yes;

112\$	EXIT:	L4con1;
113\$	COMBINE:	10,First;
114\$	PICKQ,	cyc:
		L4m1:
		L4m2:
		L4m3;
L4m1	QUEUE,	L4m1q;
115\$	SEIZE,	1:
		L4m1r,1;
137\$	DELAY:	normal(41.9,4.23,10);
117\$	RELEASE:	L4m1r,1:NEXT(91\$);
L4m2	QUEUE,	L4m2q;
116\$	SEIZE,	1:
		L4m2r,1;
138\$	DELAY:	normal(41.9,4.23,10);
118\$	RELEASE:	L4m2r,1:NEXT(91\$);
L4m3	QUEUE,	L4m3q;
135\$	SEIZE,	1:
		L4m3r,1;
139\$	DELAY:	normal(41.9,4.23,10);
136\$	RELEASE:	L4m3r,1:NEXT(91\$);
143\$	SCAN:	nq(L4m1q)==0:NEXT(112\$);

Appendix C.2

PROJECT, ;

ATTRIBUTES:

timein1:

i:

k:

l:

n;

VARIABLES:L1batchsize,1:

L2batchsize,1:

L3batchsize,1:

L4batchsize,1:

L1pan(8),1.69,1.75,1.65,1.80,1.61,1.73,1.54,1.77:

L2pan(8),1.69,1.75,1.65,1.80,1.61,1.73,1.54,1.77:

L3pan(8),1.69,1.75,1.65,1.80,1.61,1.73,1.54,1.77:

L4pan(8),1.69,1.75,1.65,1.80,1.61,1.73,1.54,1.77;

QUEUES:

L1q1,FIFO:	L2q1,FIFO:	L3q1,FIFO:	L4q1,FIFO:	L1con1q,FIFO:	L2m1q,FIFO:
L1q2,FIFO:	L2q2,FIFO:	L3q2,FIFO:	L4q2,FIFO:	L2con1q,FIFO:	L2m2q,FIFO:
L1q3,FIFO:	L2q3,FIFO:	L3q3,FIFO:	L4q3,FIFO:	L3con1q,FIFO:	L2m3q,FIFO:
L1q4,FIFO:	L2q4,FIFO:	L3q4,FIFO:	L4q4,FIFO:	L4con1q,FIFO:	L3m1q,FIFO:
L1q5,FIFO:	L2q5,FIFO:	L3q5,FIFO:	L4q5,FIFO:	L1m1q,FIFO:	L3m2q,FIFO:
L1q6,FIFO:	L2q6,FIFO:	L3q6,FIFO:	L4q6,FIFO:	L1m2q,FIFO:	L3m3q,FIFO:
L1q7,FIFO:	L2q7,FIFO:	L3q7,FIFO:	L4q7,FIFO:	L1m3q,FIFO:	L4m1q,FIFO:
L1q8,FIFO:	L2q8,FIFO:	L3q8,FIFO:	L4q8,FIFO:	L1m4q,FIFO:	L4m2q,FIFO:
					L4m3q,FIFO;

PICTURES: default;

RESOURCES:

L1r1,Capacity(1,):	L2r1,Capacity(1,):	L3r1,Capacity(1,):	L4r1,Capacity(1,):
L1r2,Capacity(1,):	L2r2,Capacity(1,):	L3r2,Capacity(1,):	L4r2,Capacity(1,):
L1r3,Capacity(1,):	L2r3,Capacity(1,):	L3r3,Capacity(1,):	L4r3,Capacity(1,):
L1r4,Capacity(1,):	L2r4,Capacity(1,):	L3r4,Capacity(1,):	L4r4,Capacity(1,):
L1r5,Capacity(1,):	L2r5,Capacity(1,):	L3r5,Capacity(1,):	L4r5,Capacity(1,):
L1r6,Capacity(1,):	L2r6,Capacity(1,):	L3r6,Capacity(1,):	L4r6,Capacity(1,):
L1r7,Capacity(1,):	L2r7,Capacity(1,):	L3r7,Capacity(1,):	L4r7,Capacity(1,):
L1r8,Capacity(1,):	L2r8,Capacity(1,):	L3r8,Capacity(1,):	L4r8,Capacity(1,):
L1m1r,Capacity(1,):	L2m1r,Capacity(1,):	L3m1r,Capacity(1,):	L4m1r,Capacity(1,):
L1m2r,Capacity(1,):	L2m2r,Capacity(1,):	L3m2r,Capacity(1,):	L4m2r,Capacity(1,):
L1m3r,Capacity(1,):	L2m3r,Capacity(1,):	L3m3r,Capacity(1,):	L4m3r,Capacity(1,):
L1m4r,Capacity(1,):			

STATIONS:

L1s1:	L2s1:	L3s1:	L4s1:	L1enter:
L1s2:	L2s2:	L3s2:	L4s2:	L1end:
L1s3:	L2s3:	L3s3:	L4s3:	L2enter:
L1s4:	L2s4:	L3s4:	L4s4:	L2end:
L1s5:	L2s5:	L3s5:	L4s5:	L3enter:
L1s6:	L2s6:	L3s6:	L4s6:	L3end:
L1s7:	L2s7:	L3s7:	L4s7:	L4enter:
L1s8:	L2s8:	L3s8:	L4s8:	L4end;

SEGMENTS: L1con1seg,L1enter,L1s1-7,L1s2-7,L1s3-7,L1s4-7,L1s5-7,L1s6-7,L1s7-7,
 L1s8-7,L1end-7:
 L2con1seg,L2enter,L2s1-7,L2s2-7,L2s3-7,L2s4-7,L2s5-7,L2s6-7,L2s7-7,
 L2s8-7,L2end-7:
 L3con1seg,L3enter,L3s1-7,L3s2-7,L3s3-7,L3s4-7,L3s5-7,L3s6-7,L3s7-7,
 L3s8-7,L3end-7:
 L4con1seg,L4enter,L4s1-7,L4s2-7,L4s3-7,L4s4-7,L4s5-7,L4s6-7,L4s7-7,
 L4s8-7,L4end-7;

CONVEYORS: L1con1,L1con1seg,5,1,Active,1,Accumulating:
 L2con1,L2con1seg,5,1,Active,1,Accumulating:
 L3con1,L3con1seg,5,1,Active,1,Accumulating:
 L4con1,L4con1seg,5,1,Active,1,Accumulating;

COUNTERS: L1jobs,,Replicate;

TALLIES: time1;

DSTATS:

nq(L1m1q):	nr(L1m1r):
nq(L1m2q):	nr(L1m2r):
nq(L1m3q):	nr(L1m3r):
nq(L1m4q):	nr(L1m4r):
nq(L2m1q):	nr(L2m1r):
nq(L2m2q):	nr(L2m2r):
nq(L2m3q):	nr(L2m3r):
nq(L3m1q):	nr(L3m1r):
nq(L3m2q):	nr(L3m2r):
nq(L3m3q):	nr(L3m3r):
nq(L4m1q):	nr(L4m1r):
nq(L4m2q):	nr(L4m2r):
nq(L4m3q):	nr(L4m3r):

REPLICATE, 5,0,86400,Yes,Yes;

SETS: L1s,L1s1,L1s2,L1s3,L1s4,L1s5,L1s6,L1s7,L1s8:
 L1q,L1q1,L1q2,L1q3,L1q4,L1q5,L1q6,L1q7,L1q8:
 L1r,L1r1,L1r2,L1r3,L1r4,L1r5,L1r6,L1r7,L1r8:
 L2s,L2s1,L2s2,L2s3,L2s4,L2s5,L2s6,L2s7,L2s8:
 L2q,L2q1,L2q2,L2q3,L2q4,L2q5,L2q6,L2q7,L2q8:
 L2r,L2r1,L2r2,L2r3,L2r4,L2r5,L2r6,L2r7,L2r8:
 L3s,L3s1,L3s2,L3s3,L3s4,L3s5,L3s6,L3s7,L3s8:
 L3q,L3q1,L3q2,L3q3,L3q4,L3q5,L3q6,L3q7,L3q8:
 L3r,L3r1,L3r2,L3r3,L3r4,L3r5,L3r6,L3r7,L3r8:
 L4s,L4s1,L4s2,L4s3,L4s4,L4s5,L4s6,L4s7,L4s8:
 L4q,L4q1,L4q2,L4q3,L4q4,L4q5,L4q6,L4q7,L4q8:
 L4r,L4r1,L4r2,L4r3,L4r4,L4r5,L4r6,L4r7,L4r8;