Spring 5-31-2000

Development of a demanufacturing system modeling and simulation tool

Jingjing Hu

New Jersey Institute of Technology

ABSTRACT

DEVELOPMENT OF A DEMANUFACTURING SYSTEM MODELING AND SIMULATION TOOL

by
Jingjing Hu

This thesis develops a demanufacturing system modeling and simulation tool as an interface between simulation methods and demanufacturers. This tool can mimic the behavior of demanufacturing facility and obtain penetrable understanding of the system. By this modeling tool, one can intuitively analyze the behavior of a system and improve the operational efficiency. Detailed designs on the simulation tool including user interface, logic, and user view are presented in this thesis. The initial prototype version of the system modeling and simulation tool has been completed with fourteen modules in the developed template. Each object in the template refers to a specific demanufacturing activity and uses detailed simulation logic behind its design to perform that activity.

The application to an electronic demanufacturing facility illustrates the usefulness of the interface to manage and improve the overall efficiency of facilities. This thesis also applies the simulation tool to a typical demanufacturing facility. Relevant system performance is analyzed. The comparison of different operational scenarios is made and the suggestion for better options of the system is discussed. This work is useful to assist designers in the design of efficient demanufacturing systems to tackle the environmental problems that may be caused by retired or faulty products.

DEVELOPMENT OF A DEMANUFACTURING SYSTEM MODELING AND SIMULATION TOOL

by
Jingjing Hu

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Department of Electrical and Computer Engineering

May 2000

APPROVAL PAGE

DEVELOPMENT OF A DEMANUFACTURING SYSTEM MODELING AND SIMULATION TOOL

Jingjing Hu

Dr. MengChu Zhou, Thesis Advisor Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Reggie J. Caudill Date
Professor of Industrial and Manufacturing Engineering and Executive Director of Multi-Lifecycle Engineering Research Center, NJIT

Dr. Edwin Hou, Committee Member Date
Associate Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Jingjing Hu

Degree: Master of Science in Computer Engineering

Date: January 2000

Undergraduate and Graduate Education

- Master of Science in Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2000.

- Bachelor of Science in Computer Science,
  Beijing Polytechnic University, Beijing , China, 1998.

Major: Computer Engineering

To my beloved family and CuLi

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. MengChu Zhou, who not only helped me as my thesis advisor, by providing valuable resources and insights, bur also gave me support and encouragement.

Special thanks to Dr. Reggie Caudill for his invaluable support.

Thanks to Dr. Edwin Hou for actively participating in my committee and for his constructive comments and recommendations.

I would like to take this opportunity to thank Ms. Lori Nanton, Ms. Lorietta Barnes and the staff at the Multi-Lifecycle Engineering Research Center (MERC) for their continuous motivation and suggestions.

Finally, I thank my friends and individuals not specifically delineated here who assisted me in this research.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF TABLES
(Continued)

Figure **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

Over the past years, the United States suffered severe environmental damage as a result of industrial growth and defense-related activities. Billions of tons of industrial waste is generated annually. Cleanup cost is estimated to be as high as a trillion dollars [1]. The damage to the environment has already affected our health and welfare. In response to the increasing public concern and growing responsibility, environmental protection becomes an important issue for industry. Pressure on the manufacturing industry to improve its environmental performance has never been stronger and is escalating daily [1, 2].

A large number of product-related environmental laws and regulations are rapidly increasing throughout the world. Topics such as product take-back and energy efficiency are receiving a great deal of attention. It is very essential to become more environmentally conscious in the development, manufacturing, use, and post-life treatment of products.

As a result of these economic and legislative restrictions, the manufacturing industry has been evaluating various options for end-of-life management. Government regulations are moving in the direction of "life-cycle accountability", the notion that a manufacturer is responsible not only for direct production impacts, but also for impacts associated with product inputs, use, transport, and disposal. Manufacturers have come to realize that end-of-life management is not a separate activity, but an integrative process beginning with the early stages of product innovation.

Electronic products like computers and household appliances have contributed significantly to the environmental burden. It is reported that the scrap electronics wastes stream could be as high as 5 to 10 million tons per year. If discarded products and waste streams can be recovered and reengineered into valuable feed streams, then sustainability can be achieved [13]. As these products become obsolete at an increasing rate, recycling processes are needed to reclaim value from the materials in the waste stream and divert them from landfills.

The search for new solutions to the increasing recycling problem, led researchers to look for different end-of-life options such as reuse, remanufacturing, reengineering and smelting to recover products, subassemblies or components and basic materials. Demanufacturing has been introduced as an integral part of product life cycle, which performs a set of functions to recover value from products and waste streams and ships these recovered materials and components for recycling/reengineering or remanufacturing [14].

Many companies have understood the importance of the relationships between the environment and their products. Indeed more and more companies are present in congresses like R'95 (recovery, recycling, and reintegration), and have integrated the parameter 'recycling' from the design of their products to the end of their life [12]. Internal demanufacturing operation has been established and several new firms have emerged specializing in demanufacturing; however these operations are facing problems of efficient and economical operations. Since disassembly issues and the post disassembly options are related to effective management and operations of demanufacturing activity, it is necessary to provide a comprehensive understanding of demanufacturing from systems perspective point of view.

1.2 Aims and Objectives

Demanufacturing is the process of dissassembling products such as electronics, automobiles, and household appliances. The parts are then sold and reused in new products, recycled, or disposed in an environmentally safe manner. From an environmental perspective, it promotes the reuse rather than the disposal of resources, helping to achieve the long-term environmental and energy recovery goals. The denamufacturing stage is very important and demands much attention from a designer during the product design stage. Indeed the time and ease of disassembly are the keys to earn or lose money in a demanufacturing process. More precisely, if a product needs a long time and many operations to be disassembled, the cost increases.

By systematically considering the different functions of a demanufacturing system and how to maximize the value recovery while minimizing residual disposal, a system simulation and modeling tool of demanufacturing facility is designed to identify opportunities to improve operational performance and overall efficiency of the system. The software tool will support all demanufacturing activities: Staging the workflow, Inspection of all collected products, Storage of products, Disassembly of products, Shredding products, and Shipping the recovered materials and components for further recovery. Thus this thesis aims to develop an integrated tool for demanufacturers to simualte the entire facility for various possible operational strategies, machine/worker utilization, material handling equipment changes, and layout planning, and to compare operational and economic trade-offs of different facility layout, equipment and operational changes. The simulation part of the tool will facilitate demanufacturers in better understanding of the system running either actual or planned.

1.3 Research Need

A present attempt is to recover and reuse components before disposing of used electronic equipment. Disposition of electronic products is raising increasingly important technical, financial, proprietary, and environmental issues. Increased environmental responsibility will generate potentials of product take back to manufacturers after disposal from user. One solution to the problem of end-of-life product management is a full service, one-stop integrated recycling facility designed to handle primary through tertiary recycling. The demanufacturing stage in its primitive stages is facing problems for its efficient operations and strategic planning. Some of the issues faced by demanufacturers are listed below [18]:

- Variation in the in-coming stream
- Condition of products
- Structure of products
- Operational strategies
- Strength of business

The need to retrieve higher profits from the demanufacturing stage has forced designers and manufacturers to consider the entire demanufacturing operation from system perspective under different conditions. These systems are complex, dynamic and stochastic. In addition, they involve large capital investments. Because of these, simulation is often used to investigate both the configuration alternatives of the system and its potential operation modes. Simulation is assisted by animation capabilities for graphical visualization of simulation outputs and presentation of results in the form of graphs and charts. The growing desire in improving productivity and reducing costs in system has resulted in a requirement of the demanufacturing system simulation tool. The simulation tool brings a science to support the demanufacturing engineers for exploring efficient operation of demanufacturing facility.

In this thesis, a methodology for simulation tool development is described and the approach developed for multiple functions of a demanufacturing system is discussed. Other approaches are reviewed and compared to the proposed methodology. Several example simulations are shown to demonstrate and compare the performance of the system.

1.4 Organization

This thesis presents a review of simulation techniques in Chapter 2. Chapter2 focuses on the modeling concept in simulation, the advantage and disadvantage of simulation, and the evolution history of simulation techniques. The modeling and simulation tool is presented in Chapter 3. Chapter 3 gives an introduction to the demanufacturing system. Detailed designs of the interface and manipulation of demanufacturing simulation tool is presented. The logic, and operand design and the implementation of the modeling tool are discussed. In Chapter 4, a case study on a typical demanufacturing company and an adoptive model is presented to illustrate software application. Analysis of the results which are collected from the simulations is presented. Serial and parallel configurations are simulated and compared. Conclusion and further research directions are presented in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to Multi-lifecycle Assessment System

Life cycle assessment (LCA) is an objective process to evaluate the environmental burdens associated with a product, process, or activity by identifying and quantifying energy and material usage and environmental releases on the environment, and to evaluate and implement opportunities to effect environmental improvements. Traditional LCA is a cradle-to-grave analysis whereas multi-lifecycle assessment emphasizes a cradle-to-cradle perspective. The multi-lifecycle approach extends the structure of traditional LCA's to include explicit consideration of demanufacturing, remanufacturing, reengineering and reuse stages. The total lifecycle engineering framework is presented in Figure 3.1. Compared to the traditional life-cycle model, which consists of four life-cycle stages: material manufacturing, product manufacturing, product use, and recovery, there are some additional definitions to Multi-lifecycle structure [7]: 1) Material production includes two stages, material extraction and materials synthesis; 2) To quantify the materials used in a packaging process, methods of transportation, distance traveled, and energy and emissions associated with these processes, a packaging and distribution stage is separated form the production stage as an independent stage; 3) The recovery and new life options of a product is the main point where MLCA differs from LCA. LCA specifies two types of recycling processes, open-loop recycling and closed-loop recycling. MLCA merges these two recycling options into one, and throughout its life form raw material extraction to final disposal. MLCA calls this stage of a product life-cycle as

Demanufacturing; 4) The reengineering stages acts as a link that closes the lifecycle loop; and 5) Remanufacturing stage is where the parts and subassemblies are reused.

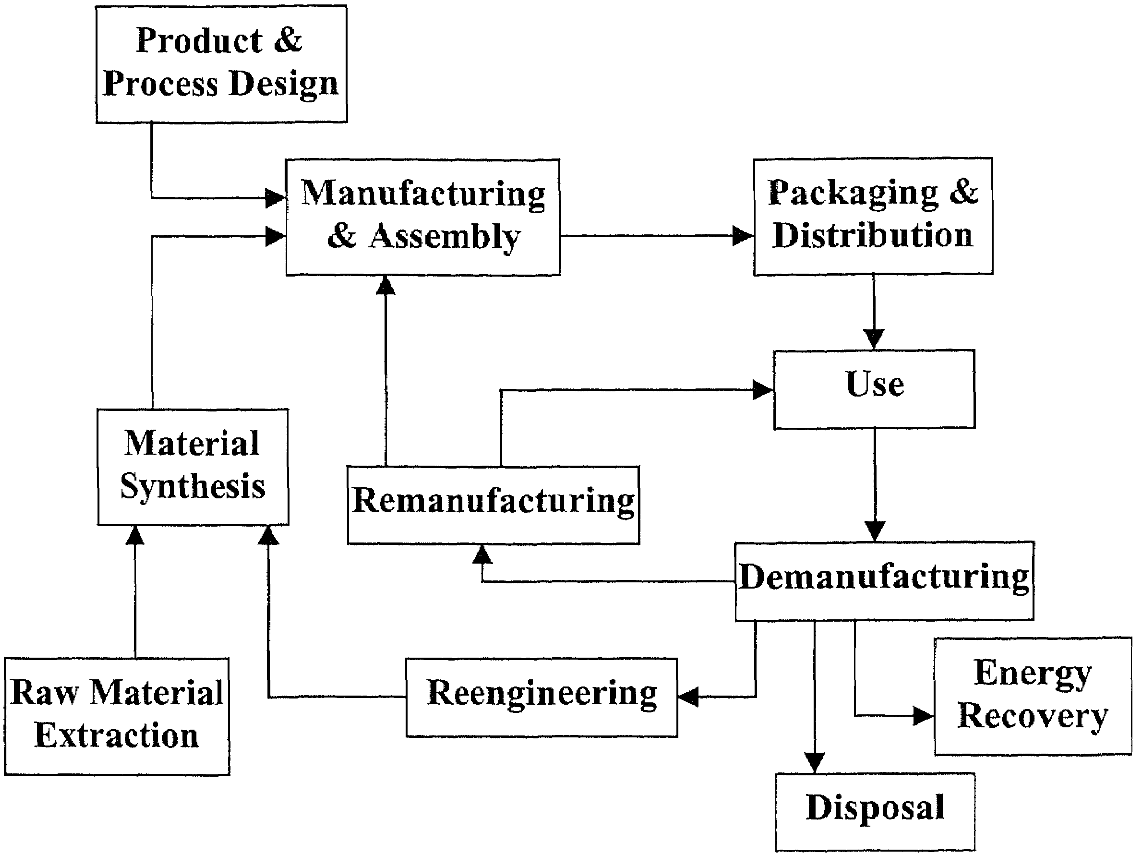


Figure 2.1 Total Life-Cycle Consideration for Analysis and Modeling [7]

Zhou, et al. [21] presents a systematic method to optimize product recovery value for electronic products. The method is used to a PC for its best design and retirement planning.

2.2 Demanufacturing System

As a very essential stage of Multi-lifecycle assessment, Demanufacturing is the process of gathering and disassembling products such as electronics, automobiles, and household appliances. The parts are then sold and reused in new products, recycled, or disposed in an environmentally safe manner. Demanufacturing also combines economic development with environmental technology to solve a dual problem. Demanufacturing provides jobs and the use of underutilized infrastructure form an urban economic development perspective. From an environmental perspective, it promotes the reuse rather than the disposal of resources, helping to achieve environmental and energy recovery goals.

Major Activities in Demanufacturing stage:

Arrival of products into facility/collection:

Discarded products and materials are collected by a demanufacturer from a variety of sources, then are separated by their own type and the recoverable value in it.

Inspection and testing: The operation helps to detect wear or assess the condition of parts subject to wear. Inspection identifies fitness of components for further use or calculates degree of wear. The process should be as easily and clearly as possible. This may involve visual check as well as few tests.

Disassembly operation: Disassembly is the process of physically separating parts and suubassemblies in a product. To recover the maximum value from the discarded products, product disassembly stage needs to be carefully analyzed in terms of sequence dependency between components and the required level of disassembly. Indeed time and ease or efforts for disassembly are key variables to earn or loose money.

Shredder and separation: A shredder is a machine primarily used for volume reduction, scrap preprocessing, pre-incineration processing and management of hazardous waste. The volume reduction facilitates handling, increases the density and lowers hauling cost. The output component of a demanufacturing process is then utilized for the next generation. The related processes are reuse, remanufacturing, reengineering and end-of-life scenario with the residue of the disassembly.

Re-use: This is the most favorable among the available options. It is the highest form of waste reduction and value recovery option. The option is justified for the serviceable products with high manufacturing cost. This can be promoted at the design stage for some products by incorporating a second use for the product [11].

Remanufacturing: Remanufacturing involves the collection of products from consumers, the transportation back to the site of manufacture, and product disassembly for the use of recovered components in the original manufacturing process. Remanufactured products are then reassembled, usually on an assembly line basis using those recovered parts and new parts where necessary. Remanufacturing then involves a rather high volume factory arrangement similar to new product manufacturing except the parts flowing to assembly lines are mostly reconditioned parts.

Reengineering: LCA addresses two types of recycling processes, open loop recycling and closed loop recycling. In open loop recycling system, a product made from virgin material is recycled into another product that is not recycled, bur disposed off, possibly after a long-term diversion, this giving only one life to the product. Closed-loop recycling occurs when a product is recycled into a product that can be recycled over and over again. The reengineering stage acts as a link that closes the lifecycle loop. Reengineering involves the characterization of waste streams and reformulation of materials derived from waste streams. Five major reengineering processes were identified for recovered materials: Reprocess, Compatibilize, Pyrolysis to Fuels, Pyrolysis/Hydrolysis to monomers, and shredding of metals. The major material inputs to this stage are obtained from demanufactured products, through process such as shredding and separation.

- Eventual disposal

Although the aim of the MLCA is to prolong the useful life of a product for as long as possible, there is still the option of disposal. It may take the form of involving thermal conversion.

- Landfill

  Landfill is a very good method of waste disposal, provided that suitable land is available within an economic distance form the collection area. It is also a good method of land recovery, although, if putrescibles (non-plastics) are present, time has to be allowed for gas generation to cease and the site to become stable. Whatever the composition of the material tipped, the site is always covered with earth before being put to useful purpose, for example as agriculture or recreational land.

- Incineration

  Incineration is a good method of treating waste as it reduces the weight and refuses volume to a minimum, leaving only metal, ash and clinker. The latter can be used for land-infill and requires only one third or less of the space needed for controlled tipping, during which crude wastes are deposited in layers into prepared excavations, or on low-lying land, compressed and covered with earth each day. The need for space can be reduced further if the ash is of sufficient quality for use as low-grade civil engineering fill. The high calorific value of most plastics means that they may serve as fuel, and in that respect may be looked upon as a resource as the sale of the energy can partly off-set the financial cost of this waste treatment. Many municipal incinerators are available, some of which are equipped to provide heating for houses or factories, or to raise the temperature of steam for use in the generation of electricity. Plastics available in mixed waste such as telephones, far from interfering with incineration, are an advantage because of their high calorific value.

2.3 Current Research in Demanufacturing

2.3.1 Design for Disassembly

Future legislation makes it important for products to be designed to facilitate recycling in some form when they reach the end of their lives. Areas which have been examined include the way in which products can be collected and disassembled, and the materials and parts reused, repaired or recycled. Ways also been considered to assess product designs for their ease of disassembly and potential to be recycled. Designers have focus on their attention towards the following aspects for achieving effective disassembly performance [12]:

- Make subassemblies and inseparably connected parts from the same or a compatible material. Most complex products have a mixture of different materials which often cannot be recycled together. This invariably results in a large amount of disassembly being needed in order to separate the materials and facilitate recycling. However often the quantities of separated materials are too small to be economically recycled and so costly disassembly does not result in all the products being recovered. This needs to change if the profit gained from a product is to be maximized and disassembly and recycling is to be made more attractive to companies.
- Locate parts with the highest value in easily accessible places. One of the aims of disassembly is to carry out the optimum amount of disassembly in order that the revenue gained from recycling the parts is greater than the cost of carrying out the

operations. Often the cost exceeds the revenue; this is because of the low value many materials have when they are recycled and also because disassembly is often awkward or difficult. The consideration of this aspect is to reducing waste movements during disassembly and logical structure to speed up the disassembly.

- Use more standardization of fasteners and only those fasteners, which are easy to disassemble or destroy.

Next, we list the guideline in Table 1 which could be used by designers to ensure that products are designed to meet the requirement for disassembly and recycling. The guidelines have also been classified according to three areas of the product design: materials, fasteners and connections and product structure. This table lists 25 guidelines divided between the three areas. The guidelines are not all distinct; many are actually dependent on each other. The following table gives the classified guidelines.

Table 2.1: Design Guidelines for Disassembly and Recycling (Materials) [12]

A. Materials	**Reason for guideline**
1. Minimise the number of different types of material.	Simplify the recycling process.
2. Make subassemblies and inseparably connected parts from the same or a compatible material.	Reduce the need for disassembly and sorting.
3. Mark all plastic and similar parts for ease of identification.	Many materials' value is increased by accurate identification and sorting.
4. Use materials which can be recycled.	Minimise waste; Increase the end-of-life value of the product.
5. Use recycled materials.	Stimulate the market for recyclates.
6. Ensure compatibility of ink where printing is required on plastic parts.	Maintain maximum value of recovered material.
7. Eliminate incompatible labels on plastic parts.	Avoid costly label removal or sorting operations.
8. Hazardous parts should be clearly marked and easily removed.	Rapidly eliminate parts of negative value.
B. Fasteners & Connections	
9. Minimise the number of fasteners.	Most disassembly time is fastener removal.
10. Minimise the number of fastener removal tools needed.	Tool changing costs time.
11. Fasteners should be easy to remove.	Save time in disassembly.
12. Fastening points should be easy to access.	Awkward movements slow down manual disassembly.
13. Snap-fits should be obviously located and able to be disassembled using standard tools.	Special tools may not be identified or available.
14. Try to use fasteners of material compatible with the parts connected.	Enables disassembly operations to be avoided.
15. If two parts cannot be compatible make them easy to separate.	
16. Eliminate adhesives unless compatible with both parts joined.	Many adhesives cause contamination of materials.
17. Minimise the number and length of interconnecting wires or cables used.	Flexible elements slow to remove; copper contaminates steel, etc.
18. Connections can be designed to break as an alternative to removing fasteners.	Fracture is a fast disassembly operation.
C. Product Structure	
19. Minimise the number of parts.	Reduce disassembly.
20. Make designs as modular as possible, with separation of functions.	Allows options of service, upgrade or recycle.
21. Locate unrecyclable parts in one area which can be quickly removed and discarded.	Speeds disassembly - see no.8.
22. Locate parts with the highest value in easily accessible places.	Enables partial disassembly for optimum return.
23. Design parts for stability during disassembly.	Manual disassembly is faster with a firm working base.
24. Avoid moulded-in metal inserts or reinforcements in plastic parts.	Creates the need for shredding and separation.
25. Access and break points should be made obvious.	Logical structure speeds disassembly and training.

2.3.2 Disassembly Process Planning

A very important aspect of disassembly is to find efficient disassembly process plans. A disassembly process plan (DPP) is a sequence of disassembly tasks which begins with a product to be disassembled and terminates at a desired level. The objective of disassembly process planning is to find optimal or near-optimal DPPs, which minimize the cost of disassembly or obtain the best cost/benefit ratio for disassembly. Identifying the 'best' disassembly sequence requires a systematic approach operating under a given set of objectives and constraints. Different methodologies to represent and find optimal disassembly sequences have been developed [19]. Homen de Mello and Sanderson proposed an AND/OR graph to present all feasible disassembly sequences and a heuristic-search algorithm to derive the optimal disassembly process plan [27, 28]. Based on algebraic and graph structural properties of Disassembly Petri net (DPN) can be solved by the Linear Programming (LP) technic [19]. Zussman and Zhou presents a DPN-based comprehensive framework for disassembly process planning and execution [22, 26].

In addition to generating good DPPs, there is a need to make the disassembly systems more efficient. Current disassembly systems are generally manual and labor intensive [20]. Therefore, automation of disassembly systems appears to be worth investigating. Umeda and Arai develop a vision system to automate the disassembly process [25]. Kopacel and Kronreif also suggest automation for disassembly operation of personal computers in order to minimize the total disassembly time [24].

2.3.3 Implementation of Disassembly and Demanufacturing Systems

One of the key purposes of dealing with discarded products is to maximize recycled resources with the consideration of ecological, environmental and economic factors. The Disassembly system should be feasible with respect to criteria on cost, energy use and environmental burden. Economic feasibility requires that within given rules and regulations, agreements and licenses, the total value of recovery should be as high as possible. After an extensive research on disassembly environment, Jovane et. al. suggest that the future research should concentrate on demanufacturing system design and developing detail methodologies for [23]:

- Configuration of manual and automated disassembly facilities
- Economic justification of designed disassembly systems and operational strategies
- Organization of a logistic network for various recovery options

2.3.4 Motivation of the System Modeling and Simulation

The increasing adoption of flexible demanufacturing systems and growing complexity of these systems have led to simulation becoming much more widely known, accepted, utilized. The industrial use of simulation consistently increases, driven by the need to understand more complex demanufacturing systems. By reproducing the activities of a system, a simulation model can provide insight into the system's overall behavior and performance potential. It can address a wide range of questions, giving design engineers an opportunity to be creative and innovative. Every demanufacturing system exhibits much the same characteristics, although differing in detail. Because of this similarity, simulation models of these systems have more common features. The differences between them will reflect the different ways in which facilities use them to form a particular system [10, 15, 16].

2.3.5 Demanufacturing Issues Addressed by Simulation

Simulation models typically have the stochastic components that mimic the probabilistic nature of a system under consideration. Successful modeling requires a close match between the collected information and the true underlying probabilistic mechanism associated with the system.

The following are some of the specific issues that simulation is used to address in demanufacturing:

The need for and the quantity of equipment and personnel

- Number, type, and layout of machines for a particular objective.
- Requirements for transporters, conveyors, and other support equipment.
- Location and size of inventory buffers
- Evaluation of a change in product volume or mix
- Evaluation of the effect of a new piece of equipment on an existing demanufacturing system
- Evaluation of capital investments
- Labor-requirements planning
- Number of shifts

Performance Evaluation

- Throughput analysis
- Time-in-system analysis
- Bottleneck analysis

Evaluation of operational procedures

- Scheduling
- Inventory policies
- Control strategies
- Reliability analysis
- Quantity-control policies

The following are some of performance measures commonly estimated by simulation:

- System throughput or productivity
- Time that a product spends in queues
- Queue sizes
- Timeliness of deliveries of disassembled components and materials to be recycled
- Utilization of equipment or personnel

2.4 Overview of Simulation Methods

A simulation is the imitation of the operation of a real-world process or system over time [5]. Simulation is used to describe and analyze the behavior of a system, and ask "what if" questions about the real system. Conceptual systems can also be modeled with simulation. By reporting the activities of a system, a simulation model can provide insight into the system's overall behavior and performance potential. Computer simulation methods have developed since the early 1960s and are the most commonly used tools in management science. The basic principles are simple. An analyst builds a model of the system, writes computer program which embodies the model and uses a computer to imitate the system's behavior [3]. This model usually takes the form of a set of assumptions concerning the operation of the system. These assumptions are expressed in mathematical, logical, and symbolic relationships between the entities, or objects of interest, of the system. Once developed and validated, a model can be used to investigate a wide variety of "what if" questions about the simulated system. Potential changes to the system can first be simulated in order to predict their impact on system performance. Simulation can also be used to study systems in the design stage, before such systems are built. Thus, simulation modeling can be used both as an analysis tool for predicting the effect of changes to existing systems, and as a design tool to predict the performance of new systems under varying sets of circumstances.

2.4.1 Modeling Concept

There are several concepts underlying simulation. These include a system and model, events, system state variables, entities and attributes, list processing, activities and delays, and finally the definition of discrete-event simulation [3,5,6].

System: A model is a representation of an actual system. The system is a collection of entities (e.g., people and machines) that interact together over time to accomplish one or more goals.

Model: An abstract representation of a system, usually containing logical and/or mathematical relationships which describe a system in terms of states, entities and their attributes, sets, events, activities, and delays.

Event: An event is defined as an occurrence that may change the state of the system. There are both internal and external events, respectively. External events are caused from the outside of the simulation system in the model. Internal events are caused by conditions defined in the definition and description of the system in the model.

System state variable: The system state variables are the collection of all information needed to define what is happening within the system to a sufficient level at a given point in time.

Entity: Entity represents an object that requires explicit definition. It could be any object or component in the system. An entity can be dynamic in that it "moves" through the system, or it can be static in that it serves other entities.

Attributes: Entities may have attributes that pertain to that entity alone. An attribute is a characteristic of all entities, but with a specific value that can differ from one entity to another.

Resource: A resource is an entity that provides service to dynamic entities. The resource can serve one or more dynamic entities at the same time. A dynamic entity can request one or more units of a resource.

Queues: A queue is a place to hold the entities which are waiting for an resources tied up by other entities.

Activities: Activities are procedures followed by the entities or performed on the entities. The duration of time for activities is known prior to commencement of the activity. Thus, when the duration begins, its end can be scheduled.

Delays: A delay is an indefinite duration that is caused by some combination of system conditions. When an entity joins a queue for a resource, the time that it will remain in the queue may be unknown initially since it may depend on other events that may occur.

2.4.2 Classification of Models

As useful working definition, simulation is the process of designing a mathematical or logical model of a real system and then conducting computer-based experiments with the model to describe, explain, and predict the behavior of the real system. In considering this definition, the ability to develop a "good" model is fundamental to the success of any simulation. There are many different types of symbolic models. To make model types more perceptible, the model could be classified as follows [4,5,6]:

- Prescriptive or descriptive
- Discrete or continuous
- Probabilistic or deterministic
- Static or dynamic
- Open loop or closed loop

Prescriptive/Descriptive: Within the first dichotomy, prescriptive models can be defined which are used to formulate and optimize a given problem, thereby providing the one best solution. On the other hand, descriptive models are defined which merely describe the system behavior and leave the optimization process totally in the hands of an analyst.

Discrete/Continuous: The discrete or continuous classification refers to the model variables. Continuous variables may take on the value of any real number, while discrete variables may assume only a limited, specified number of values. Particularly important in this distinction is the time variable in a simulation model. In a continuous model, the state of the system can change continuously over time; in a discrete model, change can occur only at separate points.

Probabilistic/Deterministic: When distinguishing between probabilistic and deterministic models, the focus is again on the model variables. If any random variables are present, we classify the model as a probabilistic model. Random variables must be defined by an appropriate probability function. If on the other hand, mathematical variables, variables whose values may be stated with certainty, are sufficient to describe the system behavior, the model then classified as deterministic.

Static/Dynamic: Models are classified as either static or dynamic depending on whether or not the model variables change over time.

Open loop/Closed loop: The notion of an open loop or a closed loop model is defined by the structure of the model rather than by the variables. Open loop models, which constitute the overwhelming majority of analytic and simulation models, have no provision for the output of the model to be fed back as input to modify subsequent outputs. In a closed loop model the output is fed back and compared to some desired level or goal to alter the system such that it maintains, or approximates, the desired value.

2.4.3 Advantages and Disadvantages of Simulation

The number of businesses using simulation is rapidly increasing. Many managers are realizing the benefits of utilizing simulation for more than just the one-time remodeling of a facility. Rather, due to advances in software, managers are incorporating simulation in their daily operation on an increasingly regular basis. Several advantages and disadvantages of system simulation are listed below [6,9]:

- Simulation enables the study of, and experimentation with, the internal interactions of a complex system.

- Informational, organizational and environmental changes can be simulated and the effect of these alterations on the model's behavior can be observed.
- By compressing or expanding time, simulation allows the operation to slow down or speed up, and thus the simulated activity could be thoroughly investigated.
- With the reconstruction and observation of a scenario, simulation can help manager understand why certain phenomena occur in a real system.
- The modern factory floor or service organization is very complex. It is impossible to consider all the interactions taking place in one given moment. Simulation allows to better understanding of the interactions among the variables that make up such complex systems.
- By using simulation to perform bottleneck analysis, the causes of the delays in work-in-process, information, materials, or other processes can be discovered.
- The typical cost of a simulation study is substantially less than 1% of the total amount being expended for the implementation of a design or redesign.
- By changing simulation inputs and observing the resulting outputs, valuable insight may be obtained into which variables are most important and how variables interact.

Disadvantages:

- Simulation results may be difficult to interpret. Since most simulation outputs are essentially random variables, it may be hard to determine whether an observation is a result of system interrelationships or randomness.
- Simulation modeling and analysis can be time consuming and expensive. Skimping on resources for modeling and analysis may result in a simulation model and/pr analysis that is not sufficient to the task.

- Numerous runs of a simulation model are usually required an dthis can result in high computer costs.
- The complex simulation models are costly and time consuming to build and run. It can be very difficult to establish the validity of the model.

2.5 Overview of Simulation Languages for Modeling

2.5.1 Language for Discrete Simulation

Computer simulation languages greatly facilitate the development and execution of simulations of complex real-world systems. But the elements and structure of most general-purpose computer languages, such as FORTRAN or PASCAL, are not closely aligned with those of systems. For example, these languages do not provide convenient data structures for event processing. This is an essential logical element in simulation modeling. Advance of the time variable, essential to running a simulation model, is also missing from FORTRAN and other general-purpose languages. Another important issue for the model development is to make the software more user friendly and more understandable to novice person. In order to include these functions and avoid complex code written in program, the development of simulation language is desired. Thus the motivation for the development and use of simulation languages stems from the desire to:

- Make the language easy to understand and use.
- Shorten the time needed to develop valid models
- Conveniently provide the statistical outputs necessary for decision making.

Figure 2.2 shows the a list of simulation languages developed over the past years.

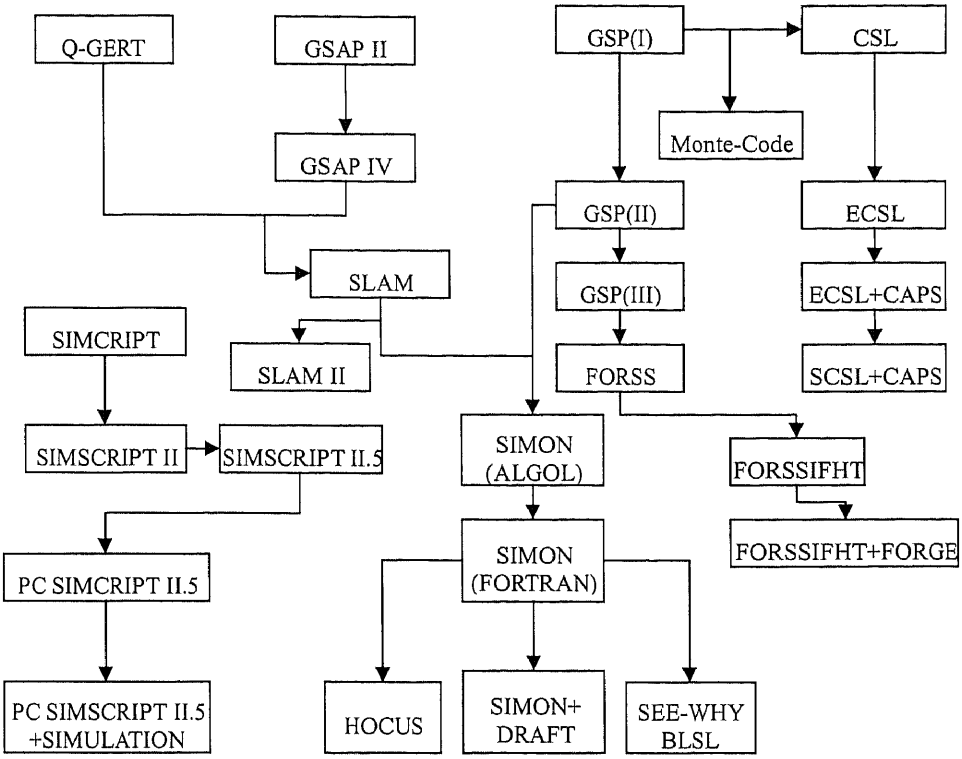


Figure 2.2 Development of Simulation Languages

The first genuine computer simulation package was written at United Steel by Dr. Tocher and his team. This package is called the General Simulation Program (GSP). It is an advanced package and has provided the basis for almost all the discrete-event languages in use today. Development continued in the iron and steel industry with GSP leading to GSP(II), GSP(III), and currently FORSS. GPSS (General Purpose Simulation System) has probably been the most widely used discrete event simulation language since 1960's. It was originally developed by Geoffrey Gordon of IBM in the 1960's, and involved through several versions, the latest being called GPSS/360 and GPSS V. Two reasons for this popularity are the ease of learning the language and the relatively short time required for building a complex model. After simulation language SIMSCRIPT was developed. A series of modifications were made for new versions of SIMCRIPT such as SIMSCRIPT II, SIMSCRIPT II.5, PC SIMSCRIPT II.5, and PC SIMANIMATION, which added a graphics capability. SIMSCRIPT was initially developed by the RAND Corporation in the 1960's, and was originally FORTRAN-based. ECSL (Extended Control and Simulation Language) is a popular British simulation language, developed by Clementson in 1960. The ECSL system and the interpreter were written in FORTRAN. SIMULA is an extension of the general purpose language ALGOL 60. Whereas ECSL is activity based, SIMULA makes use of the process. And the DEMOS package was developed to help beginners develop SIMULA programs by using a simplified structure.

Q-GERT introduced a network convention as a way of defining elemental behavior in the model and SLAM was a conceptual combination of Q-GERT. AweSim is a general-purpose simulation system. It includes the Visual SLAM simulation language to build network, subnetwork, discrete event, continuous models. General-purpose languages such as FORTRAN, ALGOL, BASIC or Pascal use an event-scheduling approach.

SIMAN (SIMulation ANalysis) is a simulation language suitable for modeling discrete-event, continuous, and combined continuous/discrete-event system. SIMAN uses a directed block diagram consisting of several block types, each of which has a unique function. SIMAN explicitly recognizes and separates three major phases of simulation analysis:

- Model definition
- Model experimentation
- Output analysis

This permits several alternative systems to be evaluated without altering the model code.

2.5.2 Simulation Language Feature

The static and dynamic structures of simulation languages provide the bare necessities for execution simulation models. Many other features provided by simulation languages are required, or highly desirable, for the effective use of simulation analysis as a decision aiding technique [9].

Model Code Development: most simulation languages still require the entering of code statements to create the model code, but the graphics capability of advanced simulation software has made possible graphical input.

Model Debugging: Once a simulation model has been coded in the selected language, the next step is to debug the code.

Random Variate Generation: For any probabilistic simulation, the ability to extract random samples from specified probability distributions is the most important.

Collection of Statistics: The collected data from the execution of simulation model is the measurement of system performance. A simulation language should permit the user to easily specify a wide variety of statistics to be collected during model execution.

Graphical Animation and Dynamic Output: The ability to use simulation languages on microcomputers permits the graphics capability of these machines to illustrate, in some fashion, the execution of the simulation model or its output.

2.5.3 Selecting a Right Simulation Language

Selecting a right simulation language is analogous to the problem of developing a right simulation model. Because the model content depends heavily on the purpose of the simulation, the language chosen depends on a particular circumstance. Beyond that, there are pragmatic issues to be considered, e.g., [6,8]:

- Ease of Learning: A flexible and easy programming language for generating complex algorithms is desired for simulation. In general, the more isomorphic the language elements are to real-world elements, the easier a language is to learn.
- Standard Code for All Computers: Initially, simulation languages were closely linked to the machine for which they were written. To run the model on another machine means significant changes. As general-purpose computer languages became more machine-independent, so do simulation languages.
- Scope of Problems Addressable by the Language: A simulation language should be able to model a wide variety of real systems. Some languages have been devised to solve a fairly small class of problems. Such languages may have the advantage of being easy to learn and use, but if the solution of a broader range of problems is ever contemplated, a more general purpose simulation language should be considered.
- Ease of understanding of the implementation: The simulation should be easy to use and understand. An animated scene of the modeled system can provide better understanding of real time status of the system and information on their states.

2.6 System Simulation Methodology

Simulation analysis is a descriptive modeling technique. As such, it does not provide the explicit problem formulation and solution steps which are an integral part of optimization models. One must specify a procedure for the development and use of simulation models in some detail to assure successful outcomes from their application. The procedure of simulation study is described as follows [2,5,6]:

1. *Program formulation.* Every study should begin with a statement of the problem. Before a problem can be formulated, it must be identified or found. Problem finding is choosing from among several problems which are competing for the same resources. Criteria for selection include technical, economic, and political feasibility, and the perceived urgency for a solution. Problem selection can have a significant impact on the ultimate success of implementation of the simulation and derivation of useful results.

   The problem formulation reveals the following required tasks:

   - Identify decision and uncontrollable variables
   - Specify constraints on the decision variables
   - Define measures of system performance and an objective function
   - Develop a preliminary model structure to interrelate the system variables and the measures of performance.

2. *Setting of objectives and overall project plan.* The objectives indicate the questions to be answered by simulation. At this point a determination should be made concerning whether simulation is an appropriate methodology for the problem as formulated and objective as stated. Assuming that it is decided that simulation is appropriate, the overall project plan should include a statement of the alternative systems to be considered, and a method for evaluating the effectiveness of these alternatives. It should also include the plans for the study in terms of the number of

people involved, the number of days required to accomplish each phase of the work with the anticipated results at the end of each stage.

3. *Model building.* Models are abstractions of the essentials of real systems. Models, however, are not right or wrong, true or false, but rather appropriate or inappropriate, sufficiently or not sufficiently accurate. In order to develop an accurate computer simulation program that implements our model, we must fully understand the system and all of its intricacies. How much detail to be included in the model of a system depends on the purpose for which the model is being developed and the marginal contribution of the additional detail.

4. *Data collection*: As the complexity of the model changes, the required data elements may also change. Also, since data collection takes such a large portion of the total time required to perform a simulation, it is necessary to begin it as early as possible, usually together with the early stages of model building. The data collection method should be tailored to the particular system setting if the data are to be collected efficiently and accurately. In selecting a method, the following issues should be considered:

   - Ability to record the data quickly enough to achieve the desired level of accuracy.
   - Impact of the data collection process on the behavior of system attribute being measured.
   - Ease of conversion of the data to computer processable form.
   - Cost of the method.

5. *Coding*: Since most real world systems result in models that require a great deal of information storage and computation, the model must be programmed for a digital

computer. With an understanding of the system and of the type of model that is to be developed. The major tasks in model construction are: 1) developing a computer program flow chart for the model; 2) selecting a programming language; 3) providing for generation of random numbers and random variant, and collection of performance measure values in the program; and 4) writing and debugging the program code.

6. *Verified*: Verification focuses on internal consistency of a model, while validation is concerned with the correspondence between the model and reality. Its pertains to the computer program prepared for the simulation model.
7. *Validated*: Validation is the determination that a model is an accurate representation of a real system. Validation is usually achieved through the calibration of the model, an iterative process of comparing the model to actual system behavior and using the discrepancies between the two, and the insights gained, to improve the model. This process is repeated until model accuracy is judged acceptable.

We relate model validation and verification to the simplified version of the modeling process in Figure 2.3.

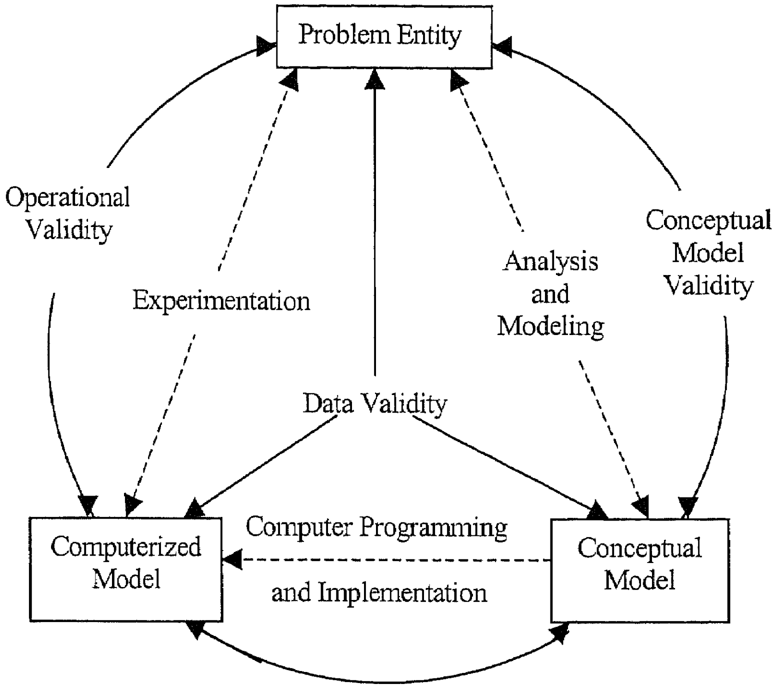


Figure 2.3 Validation and Verification Related Version of Modeling Process [8]

8. *Experimental design*: The alternatives that are to be simulated must be determined. Often, the decision concerning which alternatives to simulate may be a function of runs that have been completed and analyzed. For each system design that is simulated, decisions need to be made concerning the length of the initialization period, the length of simulation runs, and the number of replication to be made of each run.

9. *Production runs and analysis*: Production runs, and their subsequent analysis, are used to estimate measures of performance for the system designs that are being simulated.

10. *More runs?* Based on the analysis of runs that have been completed, an analyst determines if additional runs are needed and what design those additional experiments should follow.

11. *Document program and report results.* Documentation is necessary for numerous reasons. If the program is going to be used again by the same or different analysts, it may be necessary to understand how the program is operated. This enables confidence in the program so that model users and policymakers can make decisions based on the analysis. Also if the program is to be modified by the same or a different analyst, this can be greatly facilitated by adequate documentation. One experience with an inadequately documented program is usually enough to convince an analyst of the necessity of this important step.
12. *Implementation*: The success of the implementation phase depends on how well the previous 11 steps have been performed. It is also upon how thoroughly the analyst has considered the user requirements during the entire simulation process.

Figure 2.4 shows a set of steps to guide a model builder in a thorough simulation study.

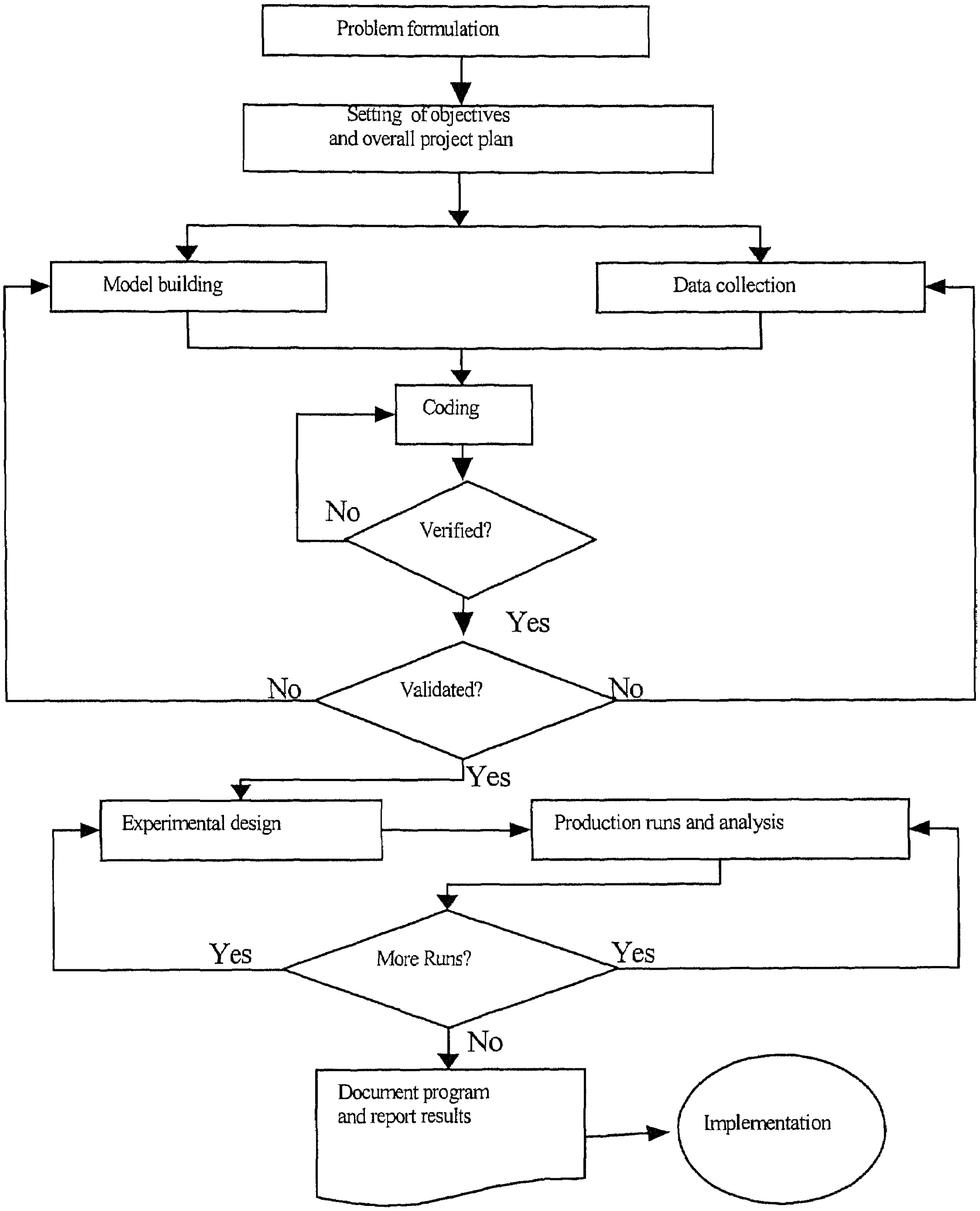


Figure 2.4 Steps in a Simulation Study

CHAPTER 3

TEMPLATE DESIGN FOR DEMANUFACTURING SIMULATION

3.1 Customization of Arena

First released in 1993, Arena software employs an object-oriented design for entirely graphical model development, and uses SIMAN as its simulation engine. To animate simulation models, Arena's core modeling structures are accompanied by CINEMA. Arena software selected as the basic software for this research is because of its object oriented approach, user defined customized templates and integration with Windows application.

The system simulation tool is developed with a demanufacturing template by utilizing the existing modules in the standard Arena template and the advanced customization functionality of Arena software. Arena application combines module structure with the flexibility of simulation languages from general-purpose procedural language like the Visual basic, FORTRAN, or C to higher level user-created templates. Figure 3.1 shows the hierarchical structure of Arena. The customized demanufacturing template utilized the modules from all abstraction levels, i.e., Common, Support, Transfer, Block and Element Panel.

A template is a panel consisting of a group of objects called modules. A module is a single construct that may be selected form a template panel and placed in a model or can be built from the combination of existing modules in other template panels [18]. The created template and all the information about modules in it are stored in *Template Files* (*.tpl), which is referred to as module definition. The information contained in the definition file is compressed to *template panel object file* (*.tpo), which is used by end users to build their own model. There are four components in a created template: model logic, operand, panel icon, and user view window.

Logic window: Logic window provides the actual logic to be performed during the simulation run. The execution and performance of the simulation is defined in this window.

Operand window: Operand is the key source that performs the user-friendly interface and provides user input information as we described in Chapter 3.

User view: This window defines the default animation feature and also the handle of a specific module. When a modeler places an instance of a module in a model window, the graphic objects that are added to the window are referred to as the user view window.

Panel icon: As an attached template, the picture of a module displayed on the template panel is referred to as the module's panel icon.

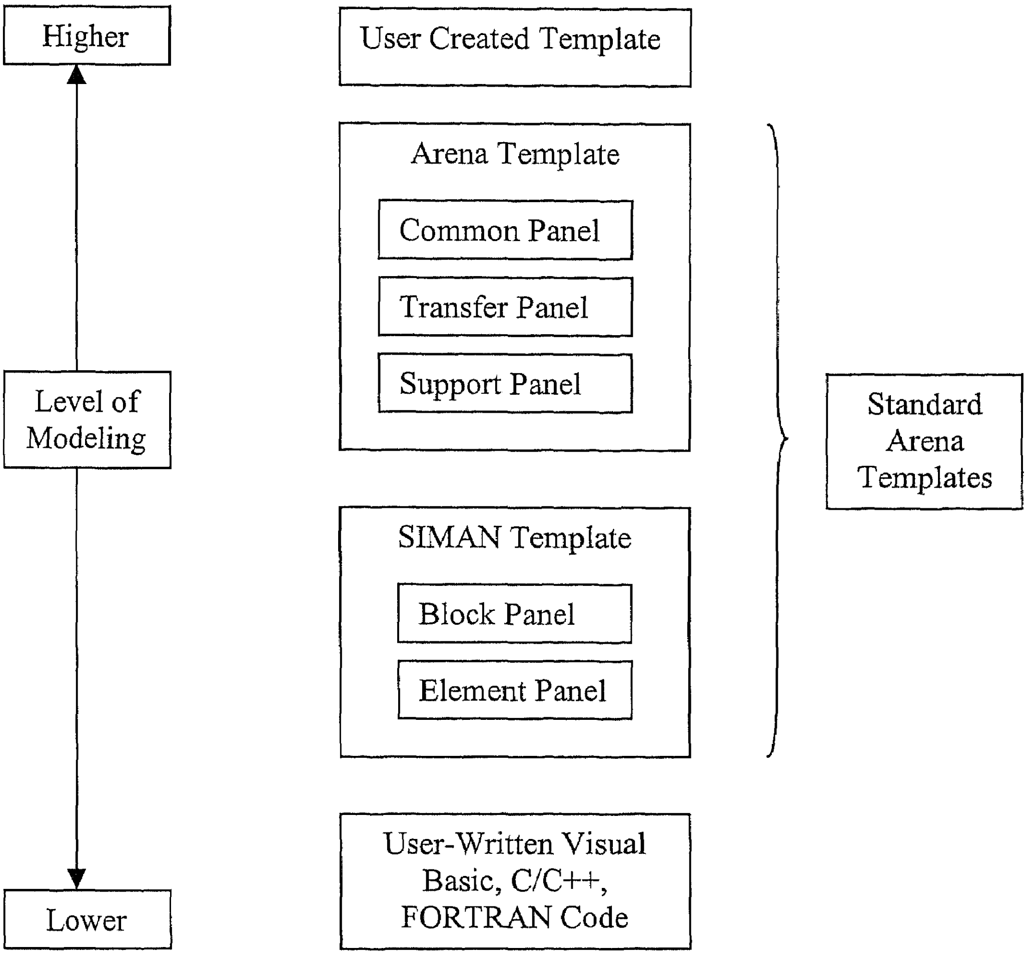


Figure 3.1 Arena's Hierarchical Structure

3.2 Design of Template

The template developed for demanufacturing simulation should include modules that perform specific demanufacturing activities and gather operational data from the simulation. The initial prototype version of a simulation tool of demanufacturing facility has been completed. It classifies the demanufacuring operation into ten activities. The modules in a developed template are corresponding to these operations. Following modules are identified for demanufacturing operation:

- Incoming Docking

- Inspection and Testing
- Storage and Staging
- Depart (Remanufacturing/Resale)
- Disassembly workstation
- Disassembled part
- Subassembled part
- Bins
- Shredder and Separator
- Shipping and Staging

The other modules are developed to improve the flexibility of the simulation tool. The corresponding functionality is described in Section 3.3. Developed modules are listed below:

- Simulate
- Transfer Device
- Distance Definition
- Signal Control

3.2.1 Incoming Docking Module Design

Figure 3.2 shows the operand window and logic window of an incoming docking module. Operand window defines the performance of the user interface and input information for the module, and has embedded connection with logic window that transfers the user input to logic module.

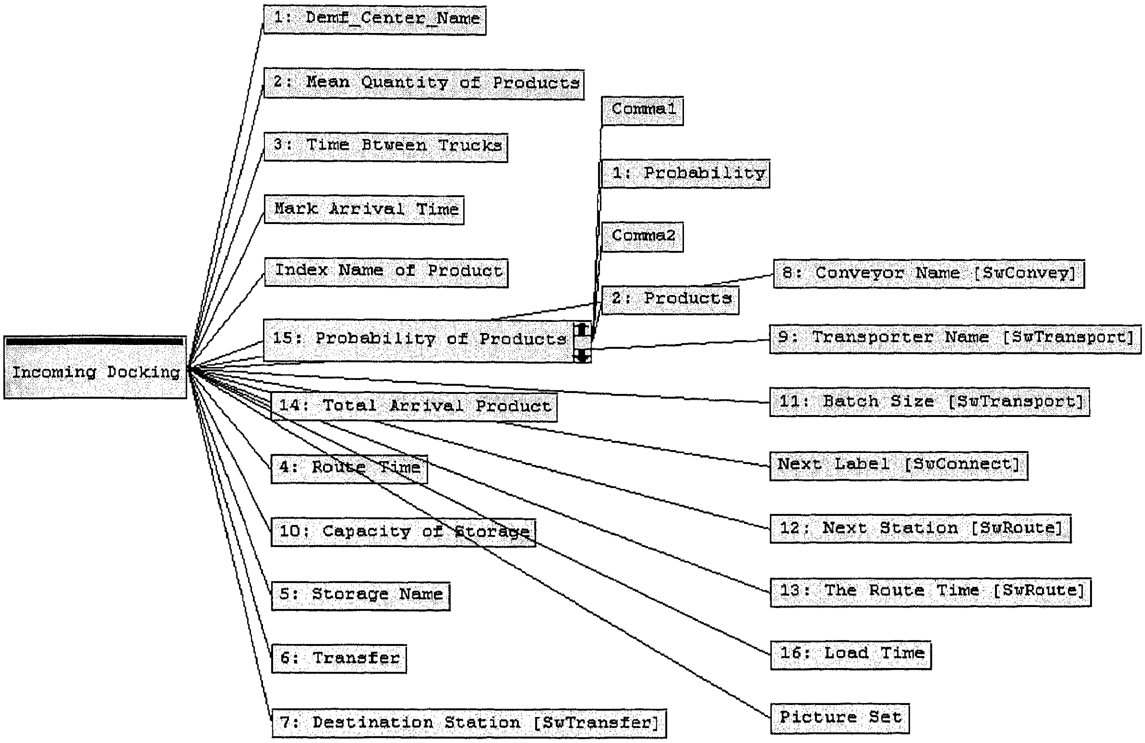

Incoming Docking
1: Demf_Center_Name
2: Mean Quantity of Products
3: Time Btween Trucks
Mark Arrival Time
Index Name of Product
15: Probability of Products
14: Total Arrival Product
4: Route Time
10: Capacity of Storage
5: Storage Name
6: Transfer
7: Destination Station [SwTransfer]
Comma1
1: Probability
Comma2
2: Products
8: Conveyor Name [SwConvey]
9: Transporter Name [SwTransport]
11: Batch Size [SwTransport]
Next Label [SwConnect]
12: Next Station [SwRoute]
13: The Route Time [SwRoute]
16: Load Time
Picture Set

Figure 3.2 (a) Operand window of Incoming Docking module

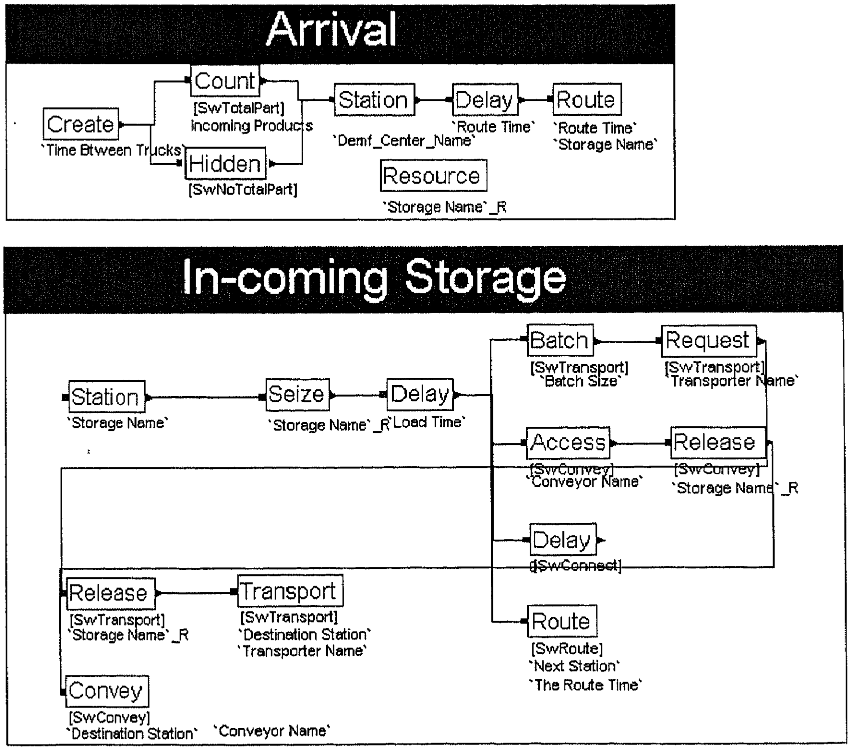


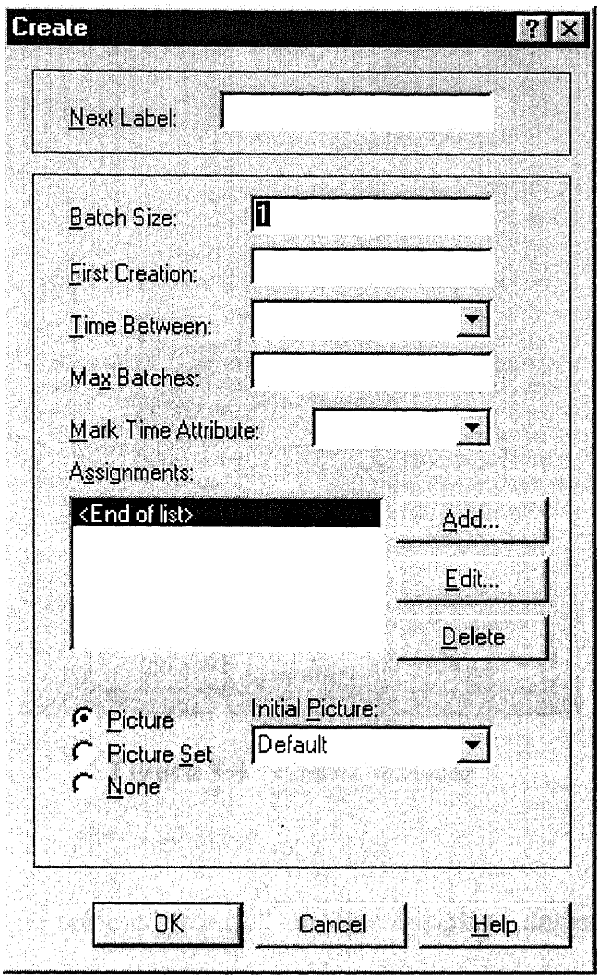
Figure 3.2 (b) Logic window

Figure 3.2 Template design of Incoming Docking module

As we described in Section 3.1, the logic window acts as a source code for execution of the intended function of a module. In this logic window we use the following embedded modules to supply the modeling tool:

- Create Module

The Create model from a Support template is the "Birth" node for arrival of entities into the model from outside by creating the modeling entities representing the various products in a demanufacturing system. Double click it to open a dialog like the one in Figure 3.3. The module is divided into two areas. In the first area, Next label describes the next module connected with this one. In the second area, Batch Size is a field defining the number of entities in each creation. First Creation describes the time triggering the operation of the whole model. Time Between defines the time between each batch creation. Mark Time Attribute gives the current simulated time an attribute name. This is typically used to track the amount of time an entity spends in all or a specific part of the model. Assignments repeat group is used to specify all attribute and variable assignments that occurs when each entity is created. Picture options determines the type, if any, of initial picture assignment for an entity when it is created.



Create

Next Label:

Batch Size: 1

First Creation:

Time Between:

Max Batches:

Mark Time Attribute:

Assignments:

<End of list>

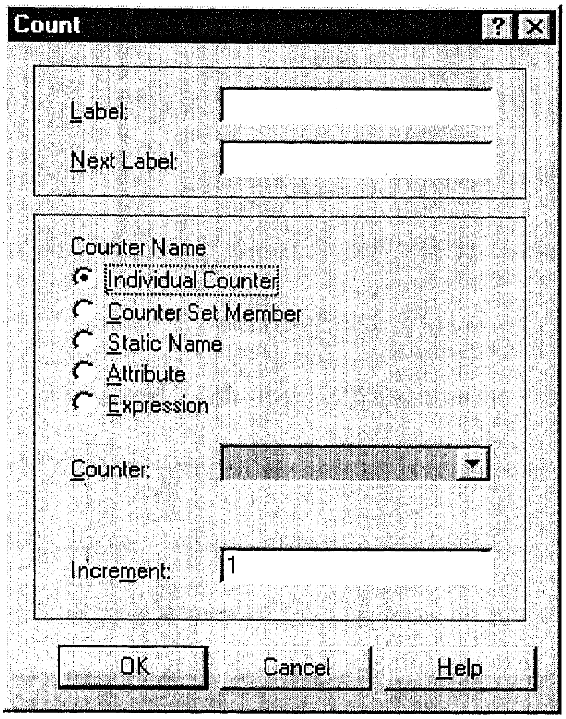
● Picture
○ Picture Set
○ None

Initial Picture: Default

Figure 3.3 Create Module

- **Count Module**

The count module from a Support template is used to calculate the entities passing through. In our incoming docking module, when an entity arrives at the Count module, the counter value is incremented by the value specified in the increment area. The final value of a Count module is the total number of products that arrive at the demanufacturing facility. Figure 3.4 represents the user screen of this module. Counter Name specifies the counter to change. It could be an Individual Counter, a set of counters, a value of static, an attribute of entity or an expression. The value may be specified as a constant or an expression that is truncated to an integer.



Count

Label:

Next Label:

Counter Name

● Individual Counter

○ Counter Set Member

○ Static Name

○ Attribute

○ Expression

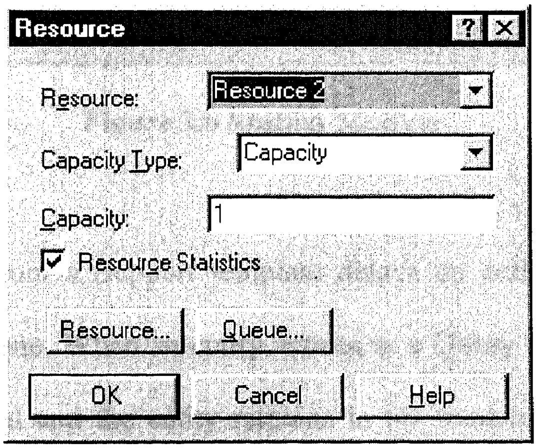
Counter:

Increment: 1

Figure 3.4 Count Module

- **Resource Module**

The Resource module selected from a Common template, defines the characteristics of the resources, including the resource names and initial capacities. Figure 3.5 shows the screen for this module. In an incoming docking module, a resource module represents the temporary storage area.



Resource

Resource: Resource 2

Capacity Type: Capacity

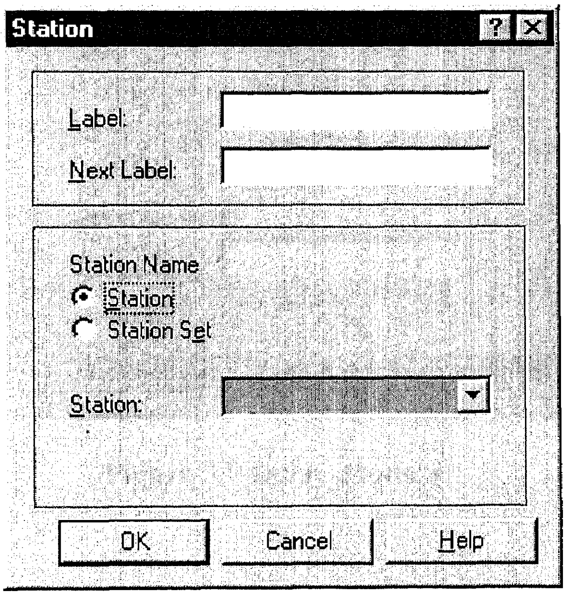
Capacity: 1

☑ Resource Statistics

Figure 3.5 Resource Module

- **Station Module**

A Station module from a Support template defines a station corresponding to a physical or logical location where processing occurs. If a Station module defines a station set, it is effectively defining multiple processing locations. Station Name determines which type of station is being defined. Station defines the symbol name of the station that is associated with this entrance point. Station modules in a demanufacturing template are presented at various locations in the system. They can be inspection workstation, disassembly workstation, or Storage and staging/warehouse, etc. The user screen of Station module is shown in Figure 3.6.



Label:

Next Label:

Station Name
● Station
○ Station Set

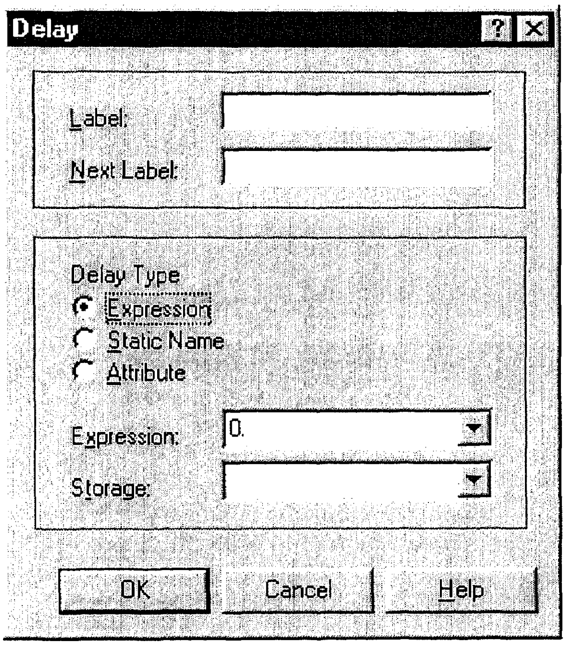
Station:

Figure 3.6 Station Module

- **Delay Module**

The Delay module from a Support template delays an entity in the system by a specified amount of time. When an entity arrives at a Delay module, the time delay expression is evaluated and the entity remains in the module for the resulting time period. The Delay Type is specified by three ways: Expression, Static Name, and

Attribute. User can input a mathematical expression whose value will be used as the entity delay time. The delay time could be Attribute which adopts user-defined entity attribute value as the delay time. Expression allows a general expression to be entered. The value of the expression is the delay time for the entity. The delayed entities are stored in the storage field. Storage field defines the storage name associated with the delay. Figure 3.7 shows the user screen for delay module.



Label:

Next Label:

Delay Type

● Expression

○ Static Name

○ Attribute

Expression: 0.

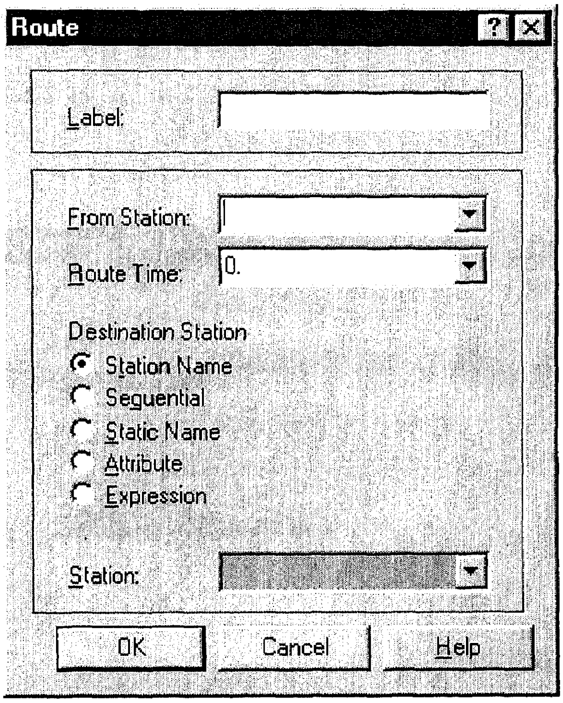
Storage:

Figure 3.7 Delay Module

- **Route Module**

A Route module is an object from a Transfer Template to transfer an entity to a specified station, or the next station in the station visitation sequence defined for the entity. A delay time to transfer to the next station may be defined. Figure 3.8 shows the Route module.

From Station defines the name of the station from which the entity is being routed. If not specified, the entity's current station location is used. Route Time describes the travel time from the entity's current location to the destination station. Destination Station defines a method for determining the entity destination location, which can be station name, sequential, static name, attribute, or expression. We use a station name option in the demanufacturing template that causes the Station text box displayed. The station field defines the name of the entity's destination station.



Label:

From Station:

Route Time: 0.

Destination Station
● Station Name
○ Sequential
○ Static Name
○ Attribute
○ Expression

Station:

Figure 3.8 Route Module

- **Seize Module**

A Seize module is in a Support template and used to allocate units of one or more resources to an entity. Figure 3.9 shows the user screen of Seize module. The Seize module may be used to seize units of a particular resource, a member of a resource set, or a resource as defined by an alternative method, such as a static or attribute. When an entity enters Seize module, it waits in a queue (if specified) until all specified resources are available.

Seize repeat group is used to define one or more resources to be allocated to an entity. If multiple resources are specified, the entity must wait in the queue (internal or external) until all the specified resource units are available. The Resource Name which appears when clicking the *add* button besides repeat group determines what type of data defines the resource to be seized. The types of a resource include Resource, Resource Set, Specific Member, Static Name, Attribute, and Expression Resource. Resource field is visible only if Resource Name is Resource. This field defines the symbol name of the resource.

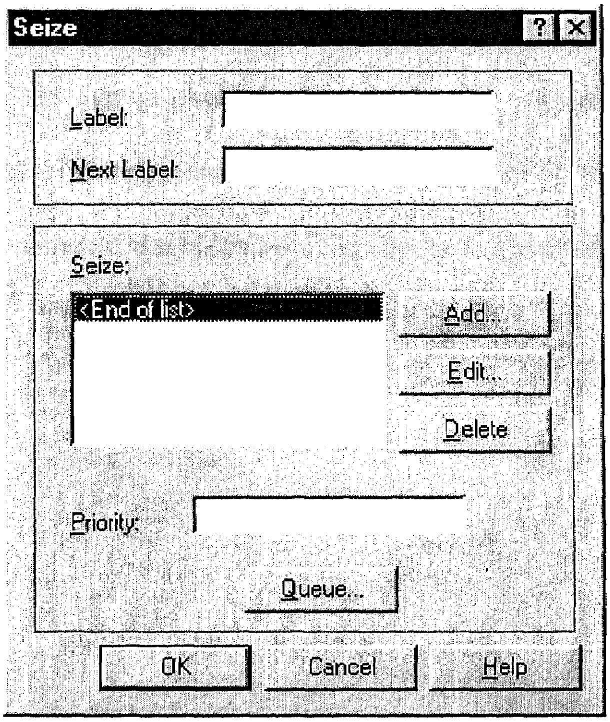
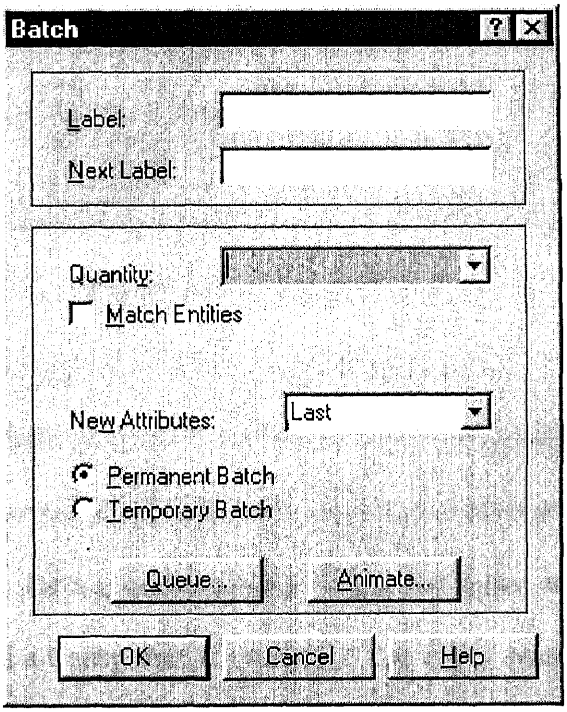


Figure 3.9 Seize Module

- **Batch Module**

A Batch module from a Support template collects entities to form a single batch for transfer activities or disassembly processing in a demanufacturing template. Figure 3.10 shows the user screen of a Batch module. Entities arriving at the Batch module are placed in a queue, if specified, until the required number of entities is reached. After this, one new representative entity is created. There are two types of batch methods: temporary batch or permanent batch. The attribute values of the batched entity are assigned values of the entities that form the batch based on the New Attributes option. When a batch is temporarily formed, the entities forming the batch are removed from the queue, and the original component entities may be recovered later using a Split module. If the entities are formed permanently, the entities forming the batch are disposed, and the original component entities are not recovered. Quantity field defines the number of entities to batch permanently or temporarily which could be used in demanufacturing template as the capacity of each transfer. New Attributes field defines the attribute assignments to the new formed batch. The Batch Method determines if the batching of the entities is permanent or temporary.



Batch

Label:

Next Label:

Quantity:

□ Match Entities

New Attributes: Last

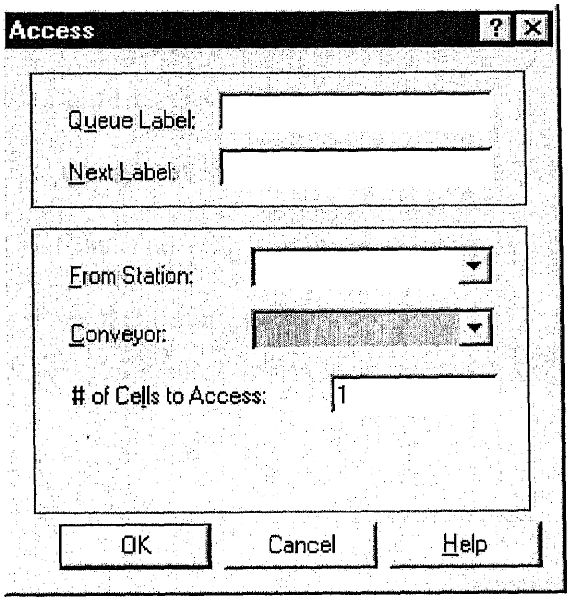
● Permanent Batch

○ Temporary Batch

Figure 3.10 Batch Module

- **Access Module**

An Access module is an object of a Transfer template used to allocate one or more cells of a conveyor to an entity for movement from one station to another. Once the entity has control of the cells on the conveyor, it may then be conveyed to the next station. From Station field in this module specifies the entity's station location. If this field is not used, the entity's current station will be used. Conveyor field defines the name of the conveyor that the entity desires.


Access

Queue Label:

Next Label:

From Station:

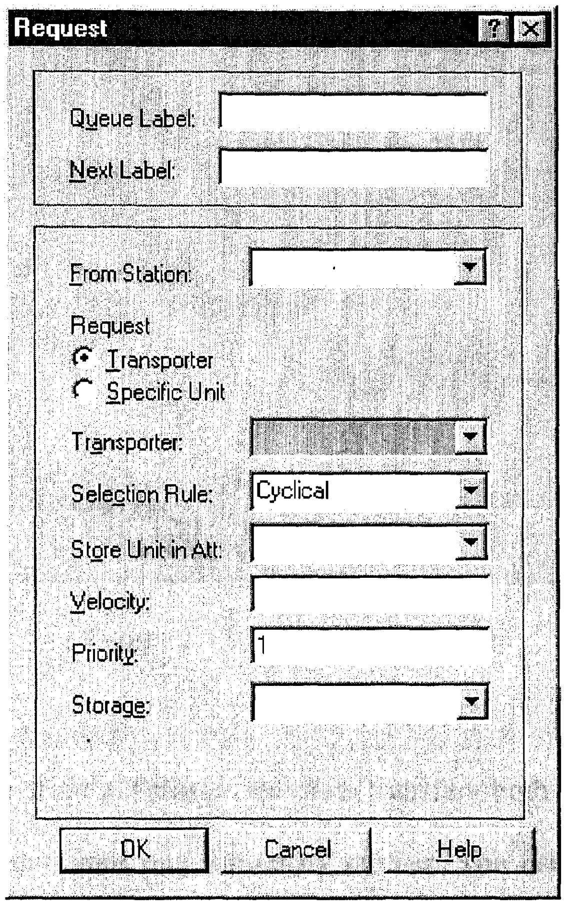
Conveyor:

# of Cells to Access: 1

Figure 3.11 Access Module

- **Request Module**

A Request module from a Transfer template is used to assign a transporter to an entity or batch and move the transporter to the entity's station location. Figure 3.12 shows the user screen for a request module. When the entity arrives at the Request module, it is allocated a transporter if available. The entity leases the request module when transporter unit reaches the entity's destination station. From Station field defines the station location of the requesting entity. If not specified, the entity's current station location is used. Transporter field defines the name of the transporter to request. If there are more than one transporter, Rule field specifies a transporter selection rule for determining which transporters to request. The selection rule has the following options: Cyclical (CYC), Random (RAN), Preferred Order (POR), Largest Distance (LDS), and Smallest Distance (SDS). Unit Number field determines the specific transporter unit in the transporter set to request. Velocity field specifies the temporary velocity at which the transporter moves toward the entity's station location. Priority field defines the priority used to request transporter units among competing modules when entities are waiting for the same transporter. Storage field defines the name of the storage at which the entity waits while the transporter is moved to the entity's station. The entity is automatically removed from the storage when the transporter arrives at the station.



Queue Label:

Next Label:

From Station:

Request

● Transporter

○ Specific Unit

Transporter:

Selection Rule: Cyclical

Store Unit in Att:

Velocity:

Priority: 1

Storage:

Figure 3.12 Request Module

- **Release Module**

A Release module is an object from a Support template used to release units of a resource that an entity previously seized. For each resource to be released, its name and quantity are specified. When the entity enters a Release module, it gives up control of the specified resource(s). Any entities waiting in queues for those resources gain control of the resources immediately. Figure 3.13 shows the user screen of a release module.

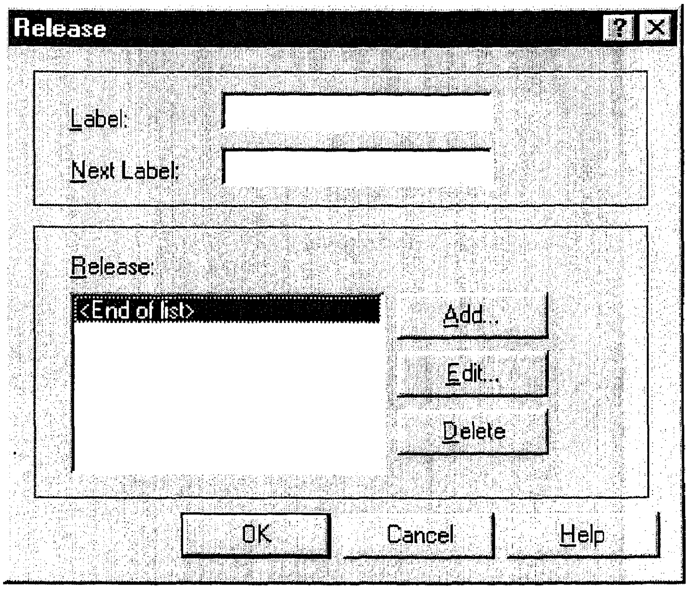
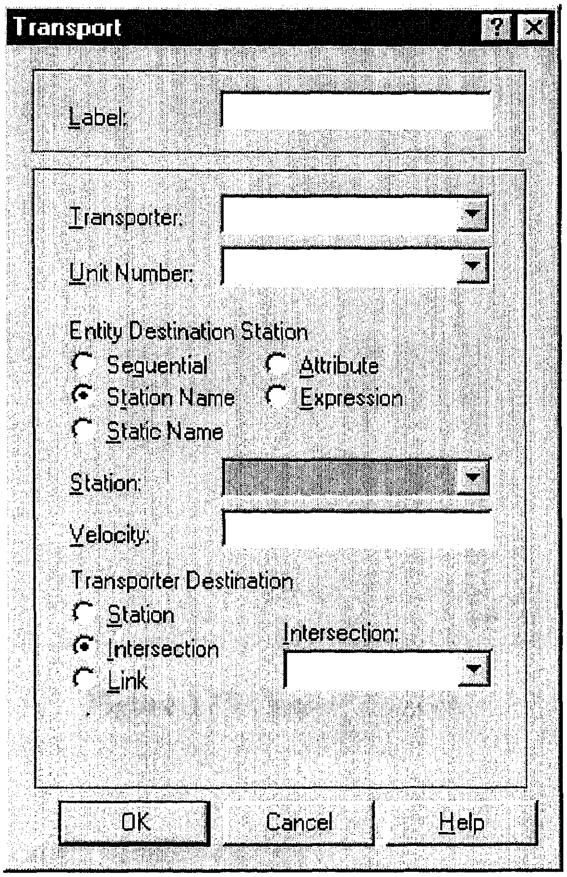


Figure 3.13 Release Module

- **Transport Module**

A Transport module from a Transfer template transfers both the controlling entity and the transporter unit from one station to another. The user screen of transport module is shown in Figure 3.14. Transport field defines the name of the transporter set to be transported. Unit Number field determines which of the transporter unit in the transporter set to transport. Destination Station determines the method for specifying the entity's destination.



Transport

Label:

Transporter:

Unit Number:

Entity Destination Station

○ Sequential ○ Attribute

● Station Name ○ Expression

○ Static Name

Station:

Velocity:

Transporter Destination

○ Station

● Intersection

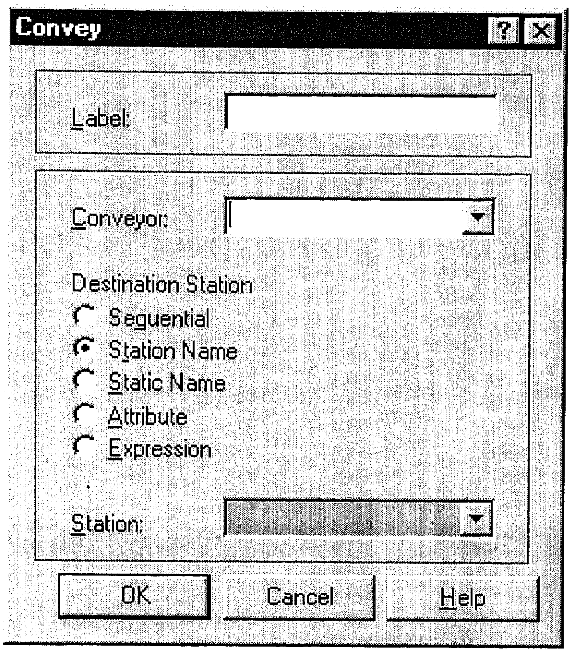
○ Link

Intersection:

Figure 3.14 Transport Module

- **Convey Module**

The Convey module moves an entity on a conveyor from its current station location to a specified destination station. The user screen of this module is shown in Figure 3.15. Conveyor field of this module defines the name of the conveyor on which the entity conveys. Station determines the method for specifying the entity destination. The time delay to convey the entity from one station to the next is based on the velocity of the conveyor (specified in the Conveyor module) and the distance between the stations (specified in the Segment module).



Label:

Conveyor:

Destination Station
○ Sequential
● Station Name
○ Static Name
○ Attribute
○ Expression

Station:

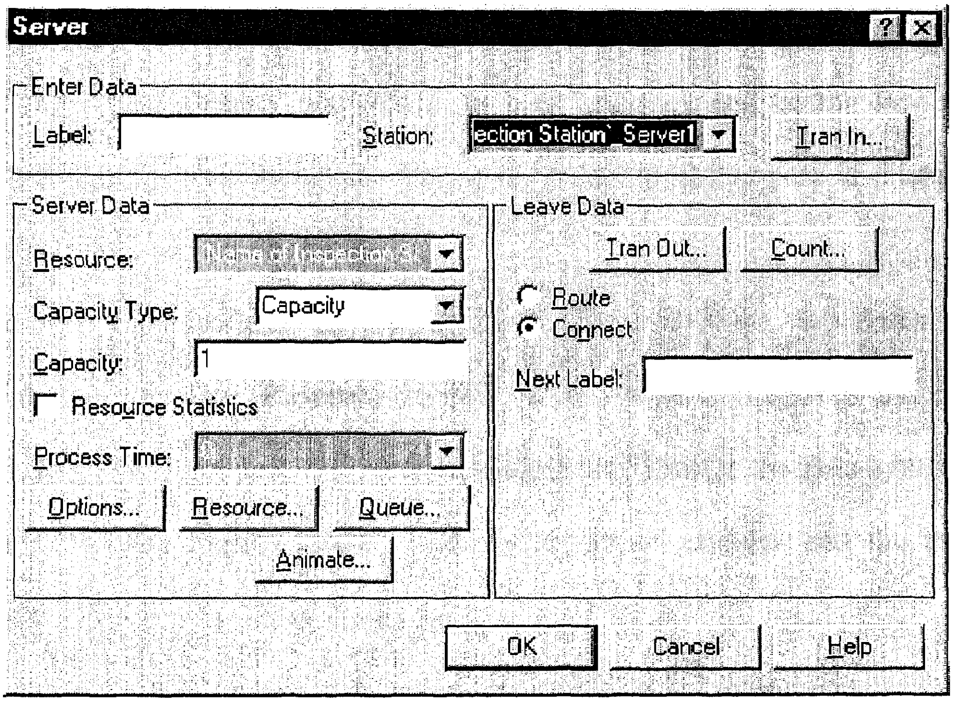
Figure 3.15 Convey Module

3.2.2 Inspection and Testing Module Design

Modules and operands used to created the Inspection and Testing module are listed below:

- Server

A Server module from a Common template is used to be an active processor. Figure 3.16 shows its user screen of server module. In a Server module, an entity enters a station, seizes a server resource, experiences a processing delay, and is transferred to another station or module. A Server module defines a station corresponding to a physical or logical location where processing occurs, i.e., disassembly workstation, and inspection workstation. An entity can be transferred to a Server module in one of three ways. First, this can happen by referencing the station associated with the Server module, then routing, conveying, or transporting to that station. Second, a graphical connection can be used to transfer an entity to a Server module. Third, an entity can be redirected to the Label associated with the Server module. The Enter Data area keeps information on Resource name and Trans-in option to free the transfer resource. The Server Data section holds the information about a processing activity. After entering the server, the entity waits in the associated queue until the server resource is available. Once the unit of resource is available, the entity seizes the server.



Label:
Station: ection Station` Server1
Resource: Name of Inspection St
Capacity Type: Capacity
Capacity: 1
□ Resource Statistics
Process Time:
○ Route
● Connect
Next Label:

Figure 3.16 Server Module

- **Split Module**

A Split module from a Support template terminates a temporary representative entity that is formed at a Batch module and recovers the original entities that form the group. The entities proceed sequentially from the Split module in the same order in which they are originally added to the group.

- **Chance Module**

A Chance module from a Support template provides branching based on the With probability rule and the Else deterministic rule. Entity destinations for each defined branch are specified by a graphical connection or by the Next Label prompt. When an entity arrives at a Chance module, it determines which probability branches, if any, will be taken based on the probabilities of the With statements specified. If the sample falls outside of the cumulative probability range representing the With probability branches, the Else branch is taken. If the Else branch has not been defined, the arriving entity is disposed. In our Inspection and testing module, this Chance module is used to specify the portion of products that pass to disassembly, remanufacturing, and resale station.

The other modules used are Station, Delay, Seize, Release, and Batch module from Support template; Resource module from Common template; and Access, Convey, Request, Transport, and Route module from Transfer template introduced in section 3.2.1. The logic window composed by these modules and the associate operand window are shown in Figure 3.17.

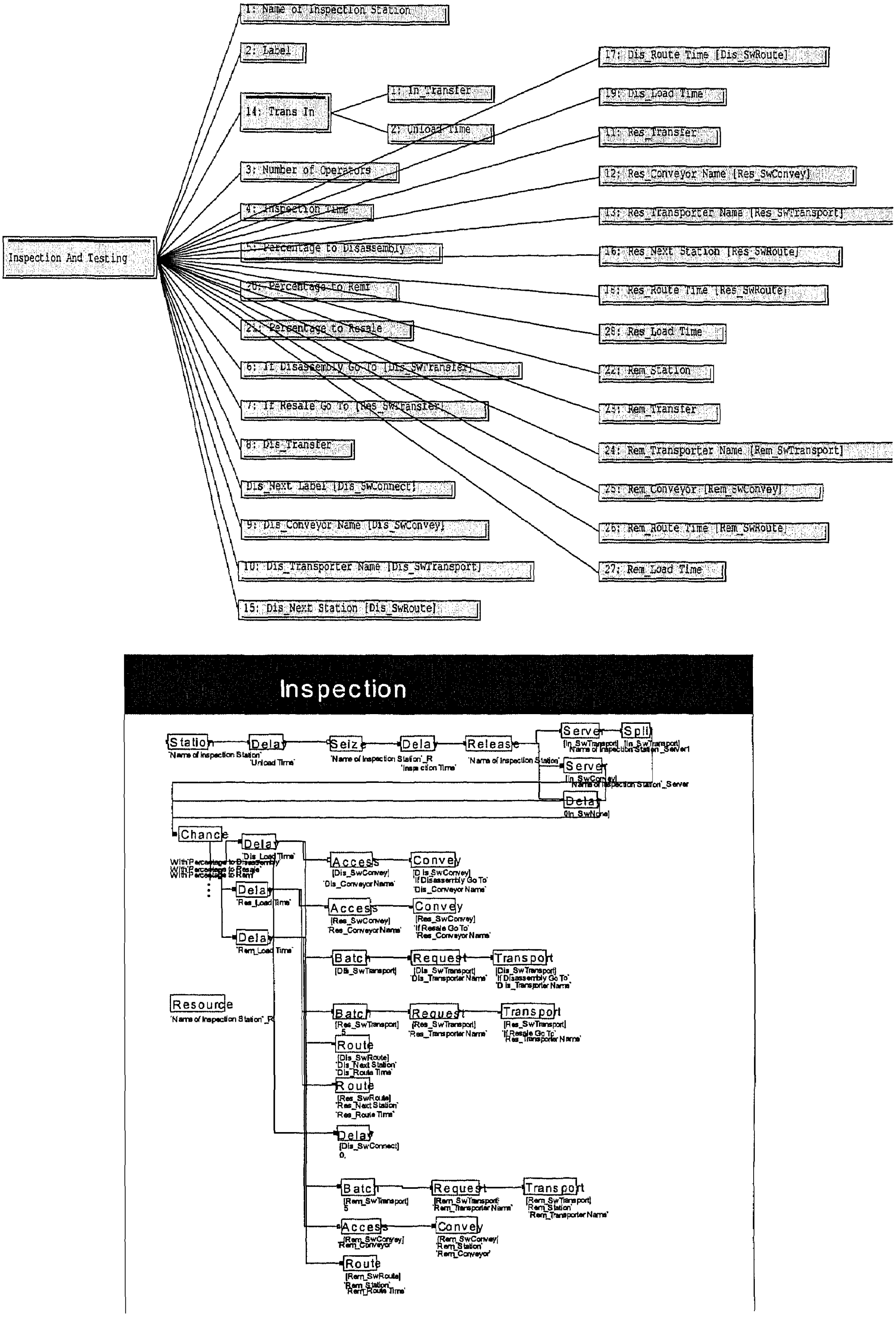


Figure 3.17 Template Design of Inspection and Testing Module

3.2.3 Storage and Staging Module Design

A storage and staging module in demanufacturing template includes five modules other than the modules we described before. They are Assign, Wait, Duplicate, Dispose form Support panel, and Variables from Common panel.

- **Assign**

  An Assign module from a Support template assigns a value to a user-defined variable, continuous rates or levels, entity attribute or picture, model status variable, or a resource state. Multiple assignments may be made by a single Assign module. When an entity arrives at an Assign module, the assignment value or state is evaluated and assigned to the variable or resource specified. If an attribute or picture is specified, the arriving entity's attribute or picture is assigned a new value.

- **Wait Module**

  Entities arriving at the Wait module are placed in a queue until a signal sent by a Signal module is received. When the signal is received, entities are released from the queue to the next module. The quantity of entities is specified in Release Limit area. The Wait module used in the storage and staging module is used to hold the products in the warehouse and wait the signal from the workstation. The signal is sent when the products in queue of disassembly workstation reach to zero. If the signal is received, which means a particular workstation desires some products, the storage and staging module releases products and transfers them to that workstation.

- **Variable**

A Variable module from a Common template specifies user-defined global variables and their initial values. Variables are referenced in the model by using their name.

- **Duplicate**

A Duplicate module from a Support template creates exact replicas of the arriving entities and sends these newly created entities to connected modules or module labels. Many modules in demanufacturing template need a duplicate module. For instance, after being processed, an entity transferred into disassembly workstation is disassembled into several same components or subssembled parts, such as screws. Thus where and how to embed the Duplicate module into the demanufacturing module is an essential issue in template development.

- **Dispose**

A Dispose module immediately disposes of an arriving entity. The disposed entity is removed from the system. This module also occurs in the depart and shipping and staging modules, which transfers the processed components out of the system.

The operand window and the logic window performing the storage and staging module are represented in Figure 3.18.

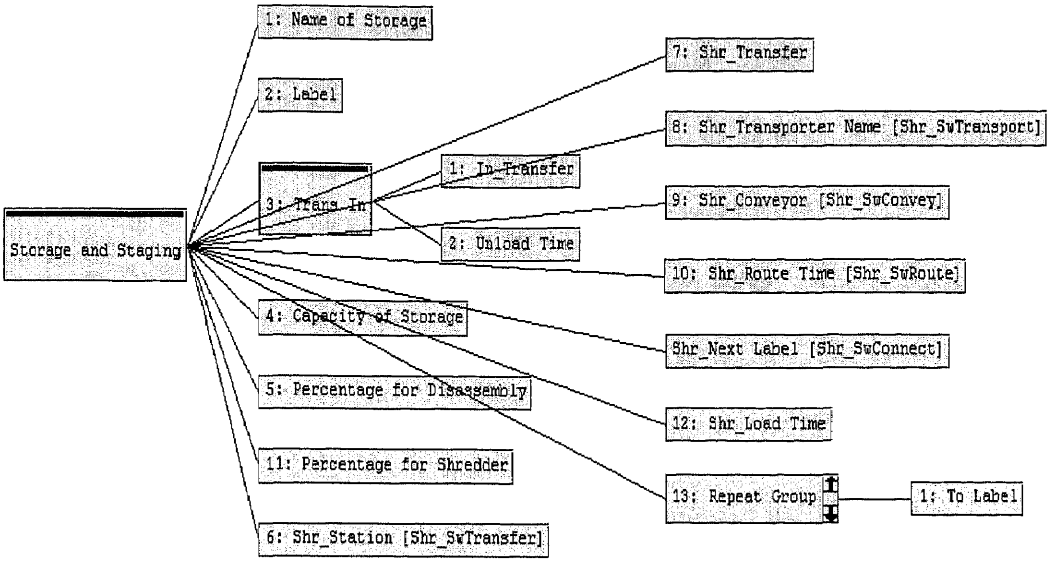

Storage and Staging
1: Name of Storage
2: Label
3: Trans In
1: In-Transfer
2: Unload Time
4: Capacity of Storage
5: Percentage for Disassembly
11: Percentage for Shredder
6: Shr_Station [Shr_SwTransfer]
7: Shr_Transfer
8: Shr_Transporter Name [Shr_SwTransport]
9: Shr_Conveyor [Shr_SwConvey]
10: Shr_Route Time [Shr_SwRoute]
Shr_Next Label [Shr_SwConnect]
12: Shr_Load Time
13: Repeat Group
1: To Label

Figure 3.18 (a) Operand Window of Storage and Staging Module

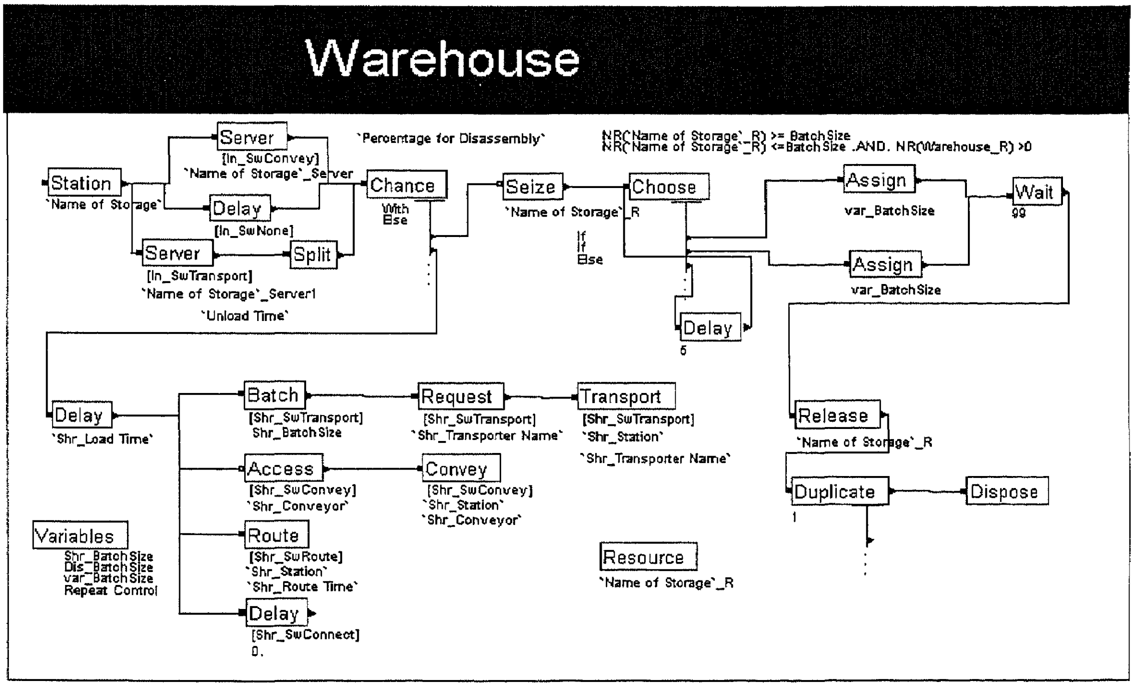


Figure 3.18 (b) Logic Window for Storage and Staging Module

Figure 3.18 Template Design for Storage and Staging Module

3.2.4 Signal Control Module Design

The Signal Control module determines the transfer mode, specifies the source of the transfer request, and makes a communicate bridge between warehouse and disassembly workstations. The modules used in the logic design include: Choose, Wait, Assign, Signal, Dispose, Delay, Batch module from Support panel; Request, Transport, Route, Access, Convey module from Transfer panel.

- **Choose**

  A Choose module provides entity branching based on the If conditional rule in conjunction with the deterministic Else and Always rules. Branch destinations are defined by graphical connections or by specifying a label destination.

  When an entity arrives at the Choose module, it examines each of the defined branch options and sends the original arriving entity to the destination of the first branch whose condition is satisfied. If no branches are taken, the arriving entity is disposed. The Choose module in a Signal Control module is the source to determine to which workstation the product is sent.

  The operand window and logic window design is shown in Figure 3.19.

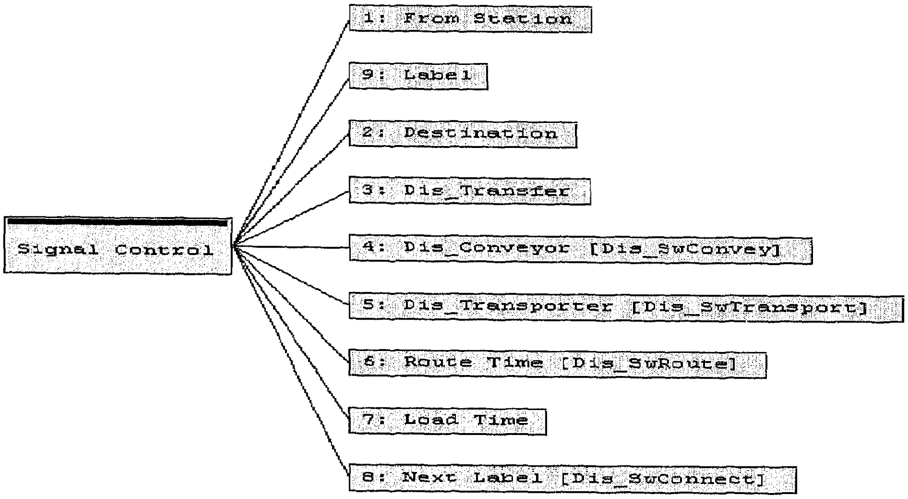

Signal Control
1: From Station
9: Label
2: Destination
3: Dis_Transfer
4: Dis_Conveyor [Dis_SwConvey]
5: Dis_Transporter [Dis_SwTransport]
6: Route Time [Dis_SwRoute]
7: Load Time
8: Next Label [Dis_SwConnect]

Figure 3.19 (a) Operand Window

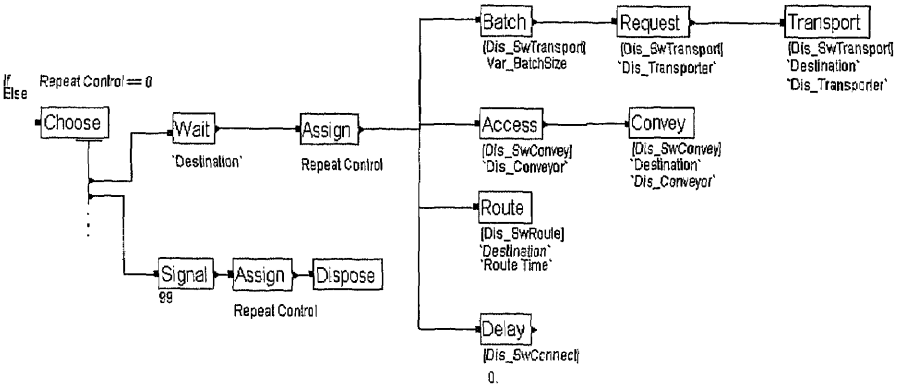


Figure 3.19 (b) Logic Window

Figure 3.19 Template Design for Signal Control Module

3.2.5 Disassembly Workstation Module Design

Disassembly workstation is the main production resource for a demanufacturing system. We use the following modules to support the disassembly capability development: Resource, Variable, Server modules from a Common panel; Station, Delay, Split, Choose, Seize, Assign, Batch, Dispose, Release, Duplicate, Create, and Signal modules from a Support panel; Request, Transport, Access, Convey, and Route modules from Transfer panel.

- **Signal**

  A Signal module sends a signal value to each Wait module in the model and releases the specified number of entities. When an entity arrives at a Signal module, the signal is evaluated and the signal code is sent. At this time, entities at Wait modules that are waiting for the same signal are removed from their queues. When a queue in a disassembly workstation module is empty, the Signal module sends signal to the Wait module in a Storage and Staging workstation and request the transfer of products.

  The operand window and logic window design is shown in Figure 3.20

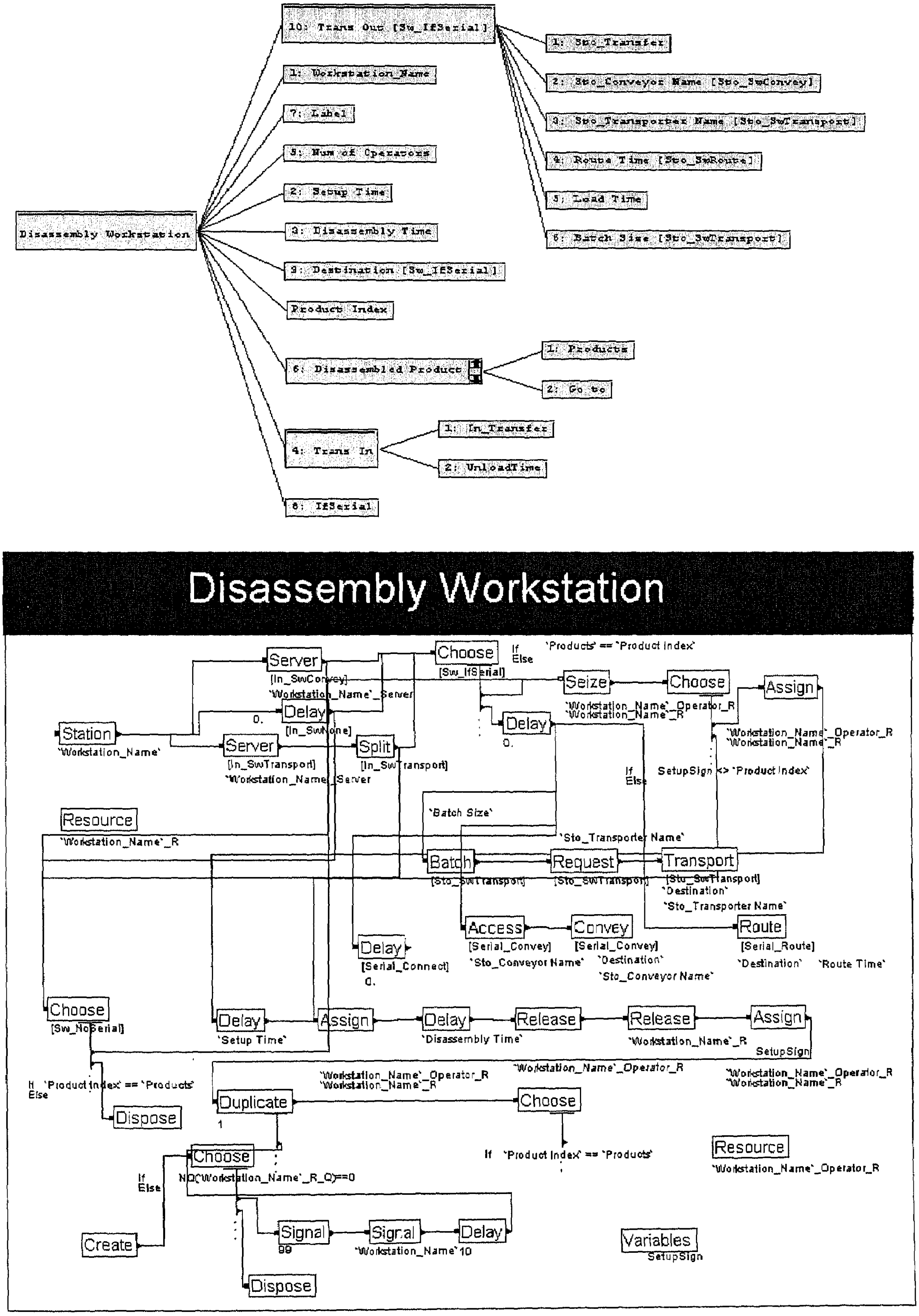
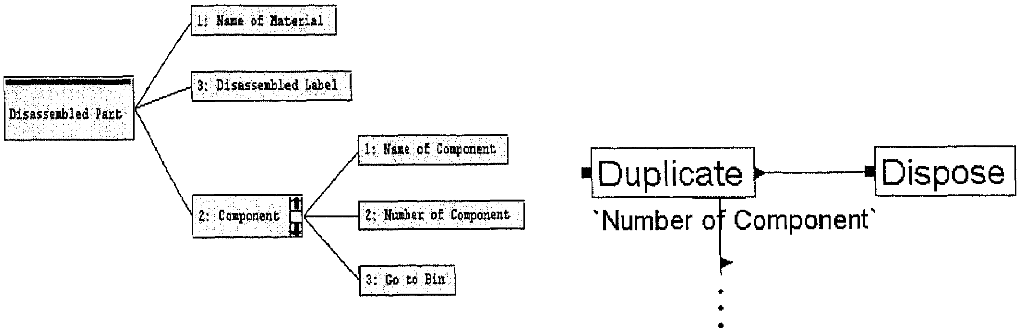


Figure 3.20 Template Design for Disassembly Workstation Window

3.2.6 Disassembled Part Module Design

Disassembly part module is designed to represent the disassembled components that need no further disassembly. The generation of components can be done using the Duplicate and Dispose module from Support template. Figure 3.21 shows the template design window. When products or subsassembled parts are transferred to disassembled part module, the embedded duplicate module generates the specified numbers of entities to present the disassembled components. Then, the dispose module discards the original entity which stand for the product being disassembled.



(a) Operand Window

(b) Logic Window

Figure 3.21 Template Design for Disassembled part Module

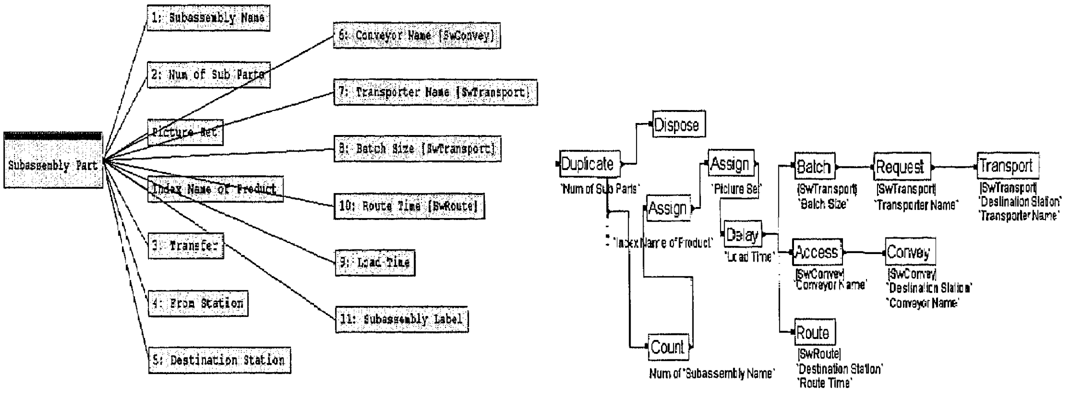
3.2.7 Subassembled Part Module Design

Subassembled part module is designed to represent the disassembled components that need further disassembly. The modules used to support the subassembly operation are: Duplicate, Dispose, Assign, Delay, Count, and Batch module from Support panel, and Request, Transport, Access, Convey, and Route modules from Transport panel.

- **Count**

A Count module increments or decrements a counter by an integer quantity. When an entity arrives at the Count module, the counter is increased or decreased by the value specified. The final value of a counter is presented as a part of the simulation output at the end of each replication.

Figure 3.21 shows the template design window.



(a) Operand Window

(b) Logic Window

Figure 3.22 Template Design for Subassembled part Module

3.2.8 Bin Module Design

A bin is a container used for collecting the disassembled parts or subassembled parts. The logic behind the Bin module is composed by: Count, Delay, and Batch modules from Support module; Request, Transport, Access, Convey, and Route modules from Transfer Panel. The template design window is shown in Figure 3.23. The count module in the logic is used to calculate the number of components that are stored in the bin. When the number of components in Bin reaches the users defined capacity. The bin asks for a transfer device to move the material in Bin to a specified storage or workstation.

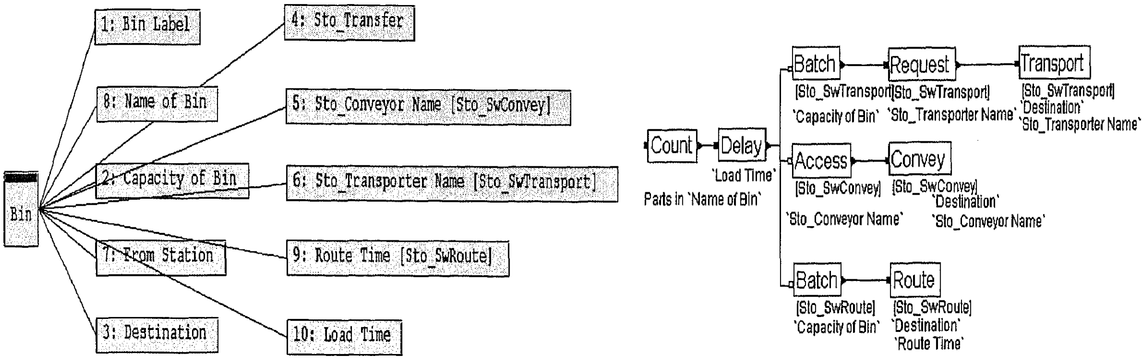


Figure 3.23 Template Design for Bin Module

3.2.9 Shredder and Separator Module Design

The Shredder and Separator modules use the following modules form arena template: Resource, and Server modules from Common panel; and Station, Split, Delay, Duplicate, Seize, Release, and Dispose from Support panel. Figure 3.24 shows the template design window for shredder and separate operation. The usage of the embedded modules is introduced in the previous Sections 3.2.1-4.

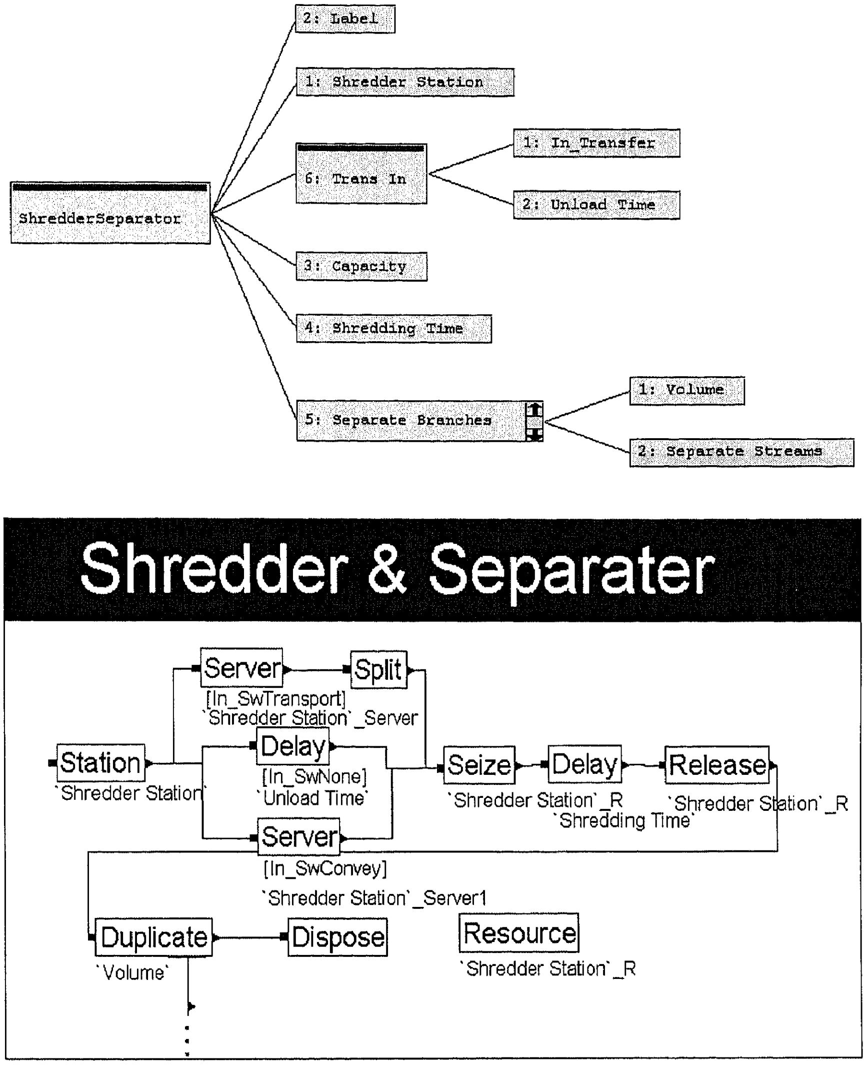


Figure 3.24 Template Design for Shredder and Separator Module

3.2.10 Shipping and Staging Module Design

Arena Modules used in the Shipping and staging logic are Resource, and Server from Common panel, Station, Delay, Split, Seize, Wait, Batch, Release, Count, Dispose, Create, and Signal modules from Support panel, and Request, Transport, and Route from Transfer panel. The template design window is shown in Figure 3.25.

- **Tally Module**

A Tally module records observations to a specified tally. The observation is defined either as an expression or one of two special functions, Interval or Between. The Interval records the time between a previous event and current simulated time for the entity arriving at the Tally. The Between type of Statistics stores the time between arrivals of entities to this Tally module. We use Tally module in logic to record the processing time of a product in a demanufacturing system.

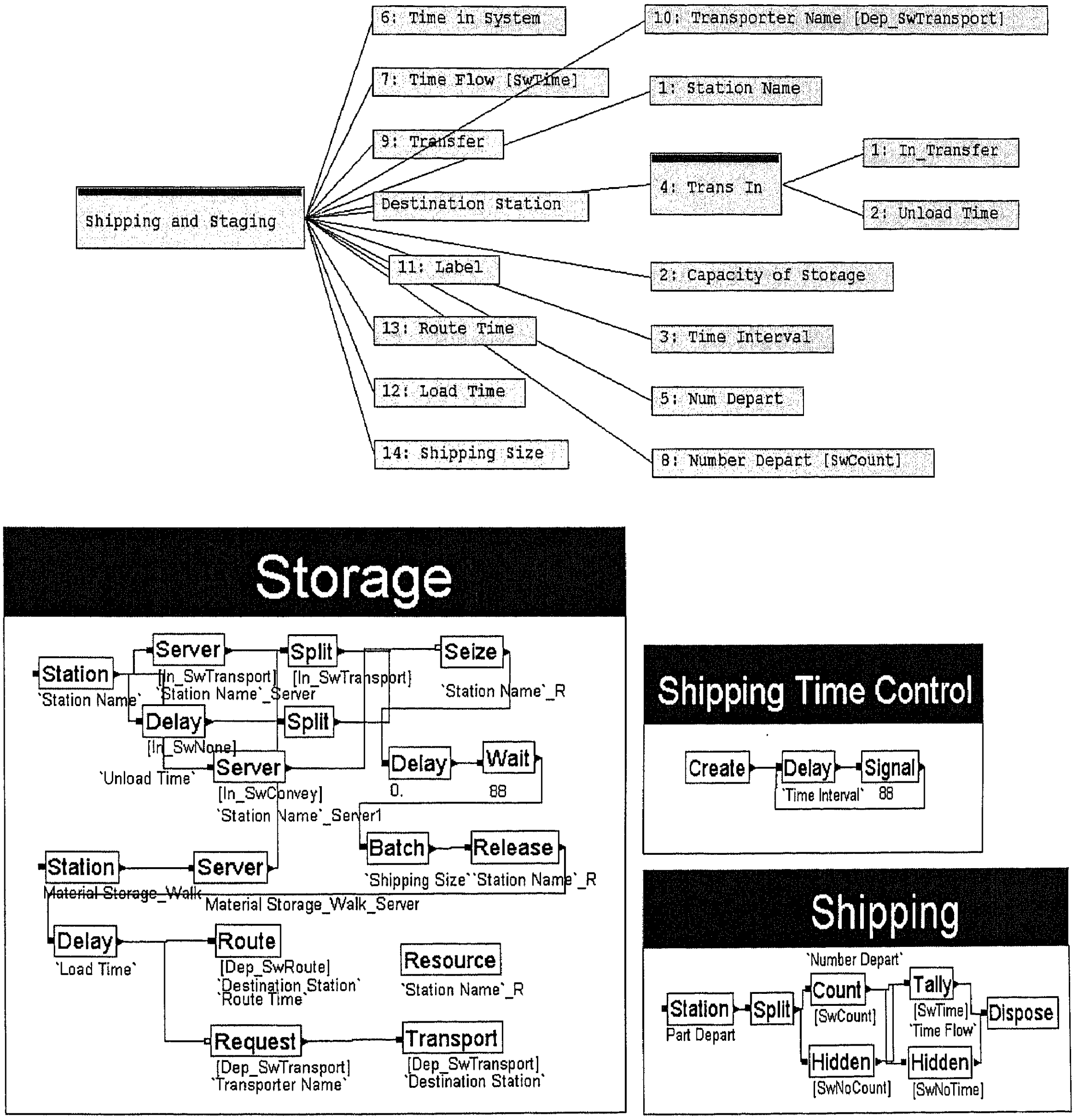
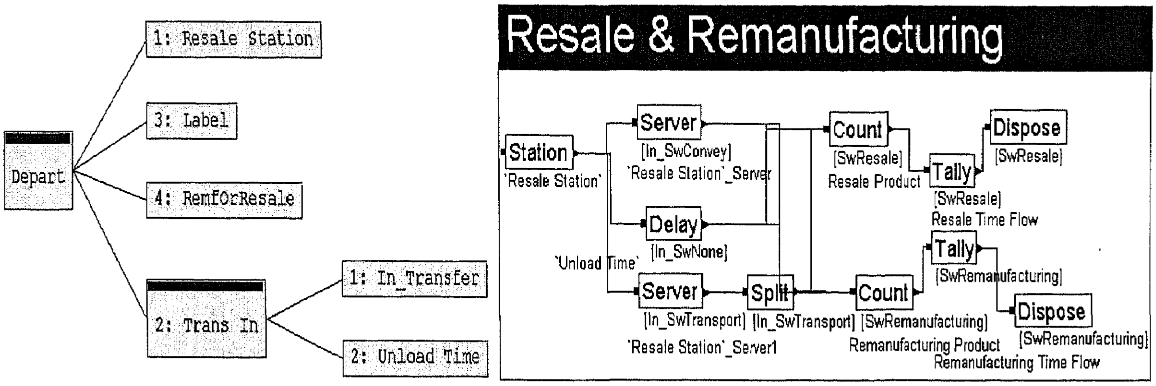


Figure 3.25 Template Design for Shipping and Staging Module

3.2.11 Depart Module Design

A Depart module uses modules Server, Station, Delay, Split, Count, Tally, Dispose, and Count in a template logic design. Figure 3.26 presents template design of a Depart module.



(a) Operand Window (b) Logic Window

Figure 3.26 Template Design of Depart Module

3.2.12 Simulation Module Design

Simulation module is designed for the products and variables definition. The modules used for the logic development are: Simulate, Variable, Expression, and Sets.

- Simulate

  The Simulate module specifies the number of simulation replications to be performed; the beginning time of the first replication; the maximum length or terminating condition for each replication; the type of initialization to be performed between replications; Finally, it allows the association of entity pictures with the simulation model.

- **Expression**

This module defines expressions and their associated values. Expressions are referenced in the model by using their names. This module used in simulation logic is to specify the processing time of each operation in a demanufacturing system, i.e. disassembly time, inspection time, or shredder time.

- **Sets**

The Sets module defines groups of similar elements that may be referenced via a common name and a set index. The elements that make up the set are referred to as the members of the set. The Sets module in a logic window is utilized to define the picture set of incoming products, subassembly parts and other objects which display on the animation screen during the execution.

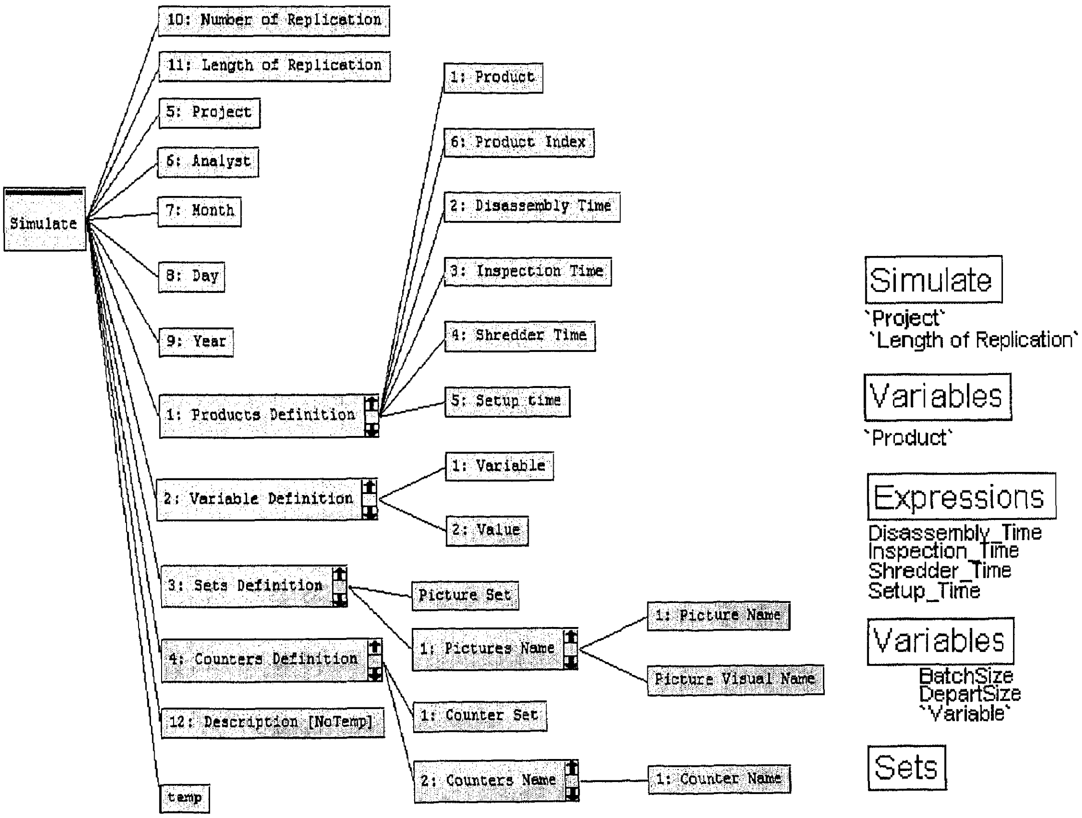


Figure 3.27 Template Design of Simulate Module

3.2.13 Transfer Device Module Design

Figure 3.28 shows the operand and logic design of a Transfer Device module. The modules of Arena template used in logic design are Transporter and Conveyor.

- **Transporter**

  A Transporter module allows the definition of a transporter device for entity movement from one station location to another. This module creates a single transporter set. Multiple sets of transporters may be defined by placing the Transporter module multiple times.

- **Conveyor**

  The Conveyor module allows the definition of either an accumulating or nonaccumulating conveyor for entity movement between stations. This module creates a single conveyor. Multiple conveyors may be defined by placing the Conveyor module multiple times.

We embedded these two modules into one module called Transfer device to make the definition easy to use. Substitute the Arena modeling terminology with a frank expression. The operand and logic design is represented in Figure 3.28.

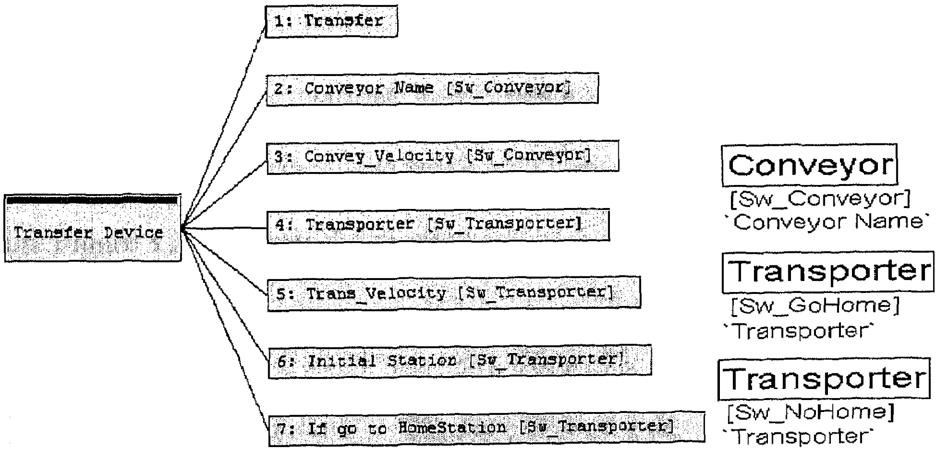

Transfer Device
1: Transfer
2: Conveyor Name [Sw_Conveyor]
3: Convey_Velocity [Sw_Conveyor]
4: Transporter [Sw_Transporter]
5: Trans_Velocity [Sw_Transporter]
6: Initial Station [Sw_Transporter]
7: If go to HomeStation [Sw_Transporter]
Conveyor
[Sw_Conveyor]
`Conveyor Name`
Transporter
[Sw_GoHome]
`Transporter`
Transporter
[Sw_NoHome]
`Transporter`

Figure 3.28 Template Design of Transfer Device Module

3.2.14 Distance Definition Module Design

Distance definition module uses the modules from Arena template like Distance, Segment, and RoutePath.

- **Distance**

  The Distance module defines the distance between two stations in the distance set. The Beginning Station, Ending Station, and Distance are used to create an appropriate distance set, which is used during the simulation run by the transporter moving between the specified stations.

- **Segment**

  The Segment module defines the distance between two stations in a segment set of a conveyor. The Beginning Station, Ending Station, and Segment are used to create an appropriate segment set, which defines the path of the conveyor.

- **RoutePath**

  The RoutePath module defines the movement path between two stations. The entity may move in either direction along the path. RoutePath modules are utilized when the transfer of an entity between two stations does not require a transporter device or a conveyor, but animation of the entity during the transfer is desired.

We embed these three modules into Distance Definition module, give it a better understandable user interface, and make a dual-direction path between the specified two station. Figure 3.29 presents the operand and logic window of template design.

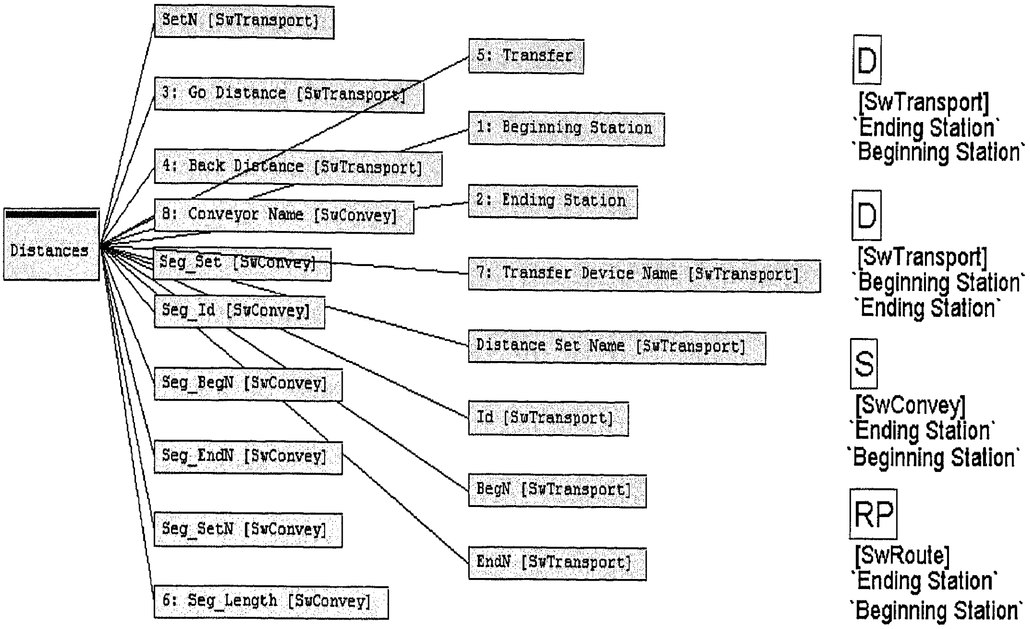


Figure 3.29 Template Design of Distance Definition Module

As the complexity of demanufacturing system under study increases, the model goes down the hierarchy to develop the customized templates. The demanufacturing system model utilizes the modules from every abstraction level to develop the customized templates. It incorporates Resource, Variable, Simulate, Expression and Sets modules from the Common template, Create, Count, Station, Delay, Seize, Batch, Release, Split, Chance, Assign, Wait, Duplicate, Dispose, Choose, Signal, Count, and Tally modules from Support template, and Route, Access, Request, Transport, Convey, Transporter, and Conveyor modules from Transfer template for modeling activities. How the developed template constructed by combining modules from different abstraction levels and the detailed information about modules utilized in the template development are presented in the previous sections 3.2.1-14.

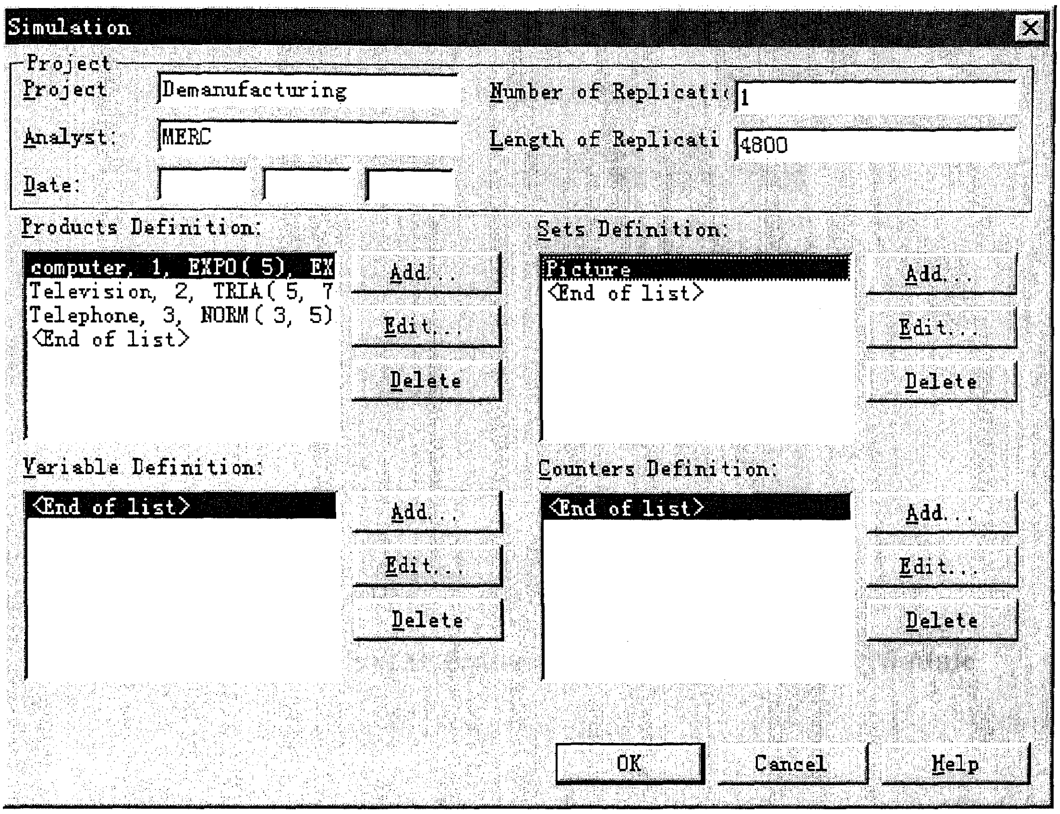
3.3 User Interface Design of the Demanufacturing Simulation Tool

Although the use of modeling and simulation in business process spreads out, it is still difficult to develop simulation models for users who do not have hard core knowledge of the particular simulation software/language. Thus, a user friendly interface between simulation knowledge and demanufacturing terminology is desired to bridge the gap and bring easy understanding of simulaition results and application of developed tools.

A simulation tool of demanufacturing facility supplies an easy windows-based interface by using the advanced Arena Customization of Template technology. The simulation tool development is classified in fourteen modules: Simulation, Incoming Storage, Inspection and Testing, Depart, Storage and Staging, Signal Control, Disassembly, Workstation, Disassembled Part, Subassembled Part, Bin, Shredder & Separator, Shipping and Staging, Transfer Device, and Distance Definition.

3.3.1 Simulation Module Interface Design

To perform a simulation on a demanufacturing process, we have to address the most important issue, i.e., how long we need to simulate it, what kind of product we deal with, and the animation presentation of the product. The simulation module interface is designed for the simulation information gathering. The interface design is shown in Figure 3.1. Each field is discussed as follows.



Simulation

Project

Project: Demanufacturing

Analyst: MERC

Date:

Number of Replicati 1

Length of Replicati 4800

Products Definition:

computer, 1, EXPO(5), EX
Television, 2, TRIA(5, 7
Telephone, 3, NORM(3, 5)
<End of list>

Sets Definition:

Picture
<End of list>

Variable Definition:

<End of list>

Counters Definition:

<End of list>

Figure 3.30 Simulation Module Screen of the Demanufacturing Template

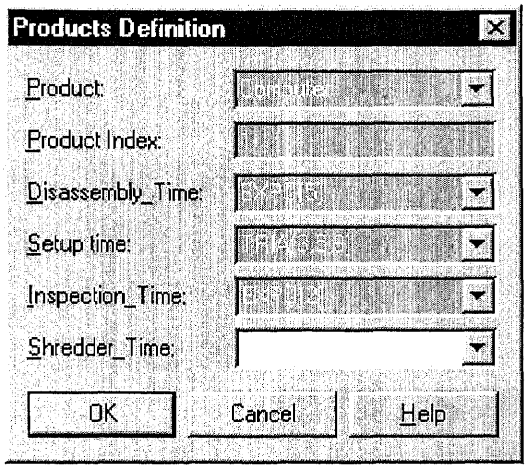
1) Project Area:

This area contains the description about a simulation project, i.e., project name, analyst name, and the date of the project developed. The most important part is Length of Replication and Number of Replications. Length of Replication is measured by minutes. Only the Number of Replications and Length of Replication are required, and all other information in this area is optional.

2) Product definition sub-window

This sub-window contains the description of all the products and subassemblies that represent in the demanufacturing process. User provides information such as Product Name, Disassembly Time, Inspection Time, Shredder Time, and Set-up time.

When user click the "Add" button inside the Production definition sub-window, the following window pops up as shown in Figure 3.2.



Product: Computer
Product Index: 1
Disassembly_Time: EXPO(5)
Setup time: TRIA(3,5,9)
Inspection_Time: EXPO(3)
Shredder_Time:

Figure 3.31 Product Definition Window of Simulation Module

Disassembly Time: Disassembly Time is the time needed of each product for the disassembly operation in a disassembly workstation.

Setup Time: The different product requires different tools for the disassembly operation. Whenever operators in the disassembly workstation disassemble a product other than the previous one that they work on, they need to change the tools. The input here is the time consumed in setting up the tools.

Inspection Time: Inspection Time is the time needed for the Inspection operation in the inspection workstation.

Shredder Time: Shredder Time is the time needed for the shredder operation in the shredder and separator workstation.

3) Set definition sub-window

Set definition sub-window contains the picture definition of product, subassembly part, and disassembled part. This definition will create the animation performance of entities in demanufacturing process. User picks the picture name from the pull-down list or fills in additional entry if the expected name does not exist in the list.

When user click the "Add" button inside Set Definition sub-window, next window pops up as shown in Figure 3.3.

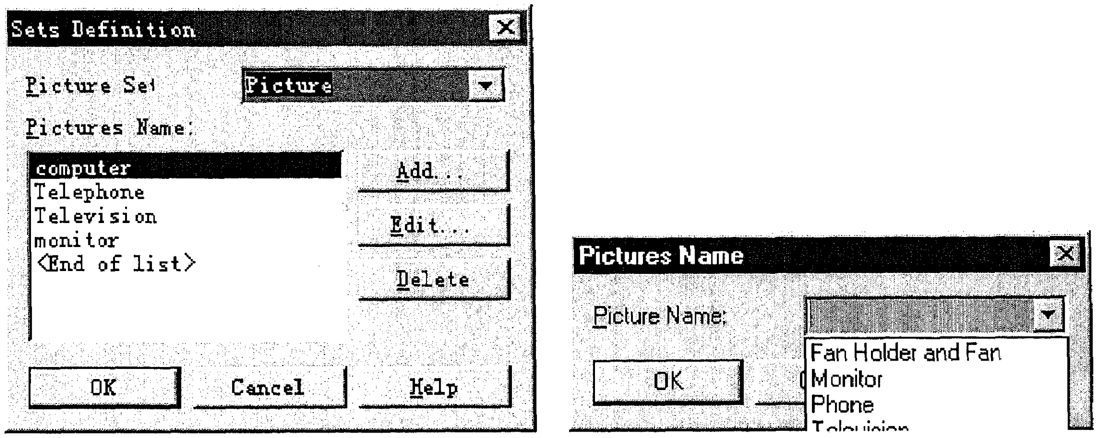
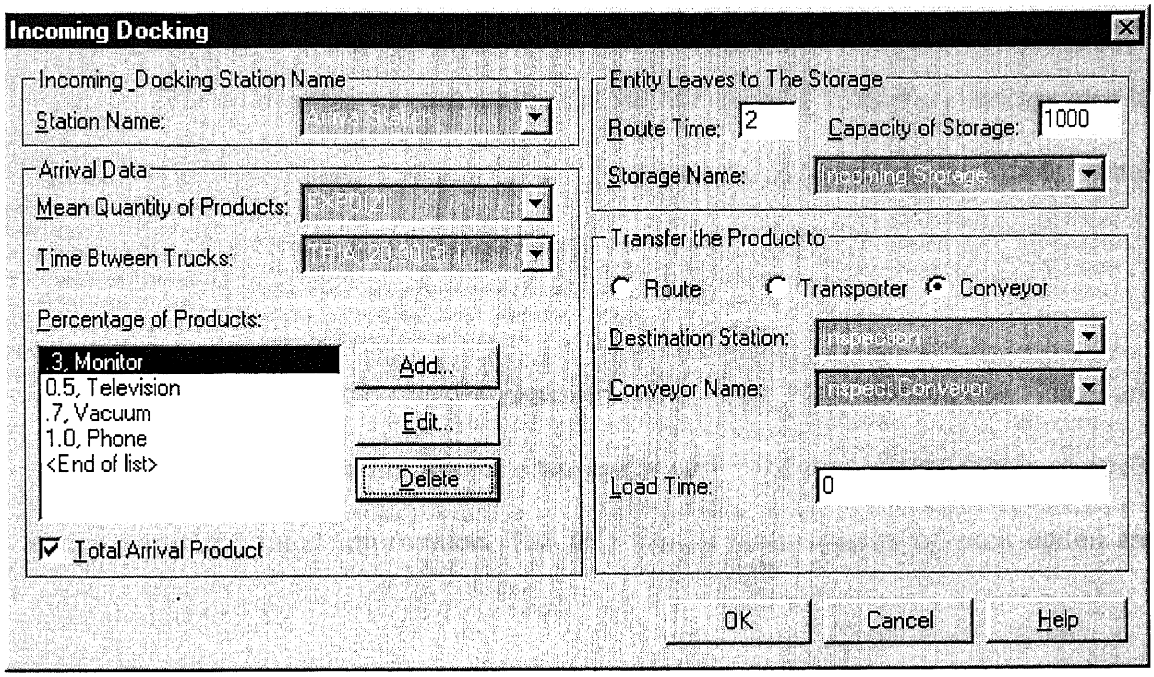


Figure 3.32 Picture Definition window of Simulation Module

After user finishes the input of this module, they can click "OK" button in the screen and save the change, or "Cancel" button to ignore the change.

3.3.2 Incoming Docking Module Interface Design

Incoming Docking module represents as the threshold of the whole facility. All the end-of-life products from households or companies are gathered in this module, then processed by the workstations in the facility. Incoming Docking module is divided into four areas: Incoming Docking Station Name, Arrival Data, Entity Leave to the Storage, and Transfer the Product to area. The interface design of this module is shown in Figure 3.4.



Incoming Docking

Incoming_Docking Station Name

Station Name: Arrival Station

Arrival Data

Mean Quantity of Products: EXPO(2)

Time Btween Trucks: TRIA(20,30,31)

Percentage of Products:

.3, Monitor
0.5, Television
.7, Vacuum
1.0, Phone
<End of list>

☑ Total Arrival Product

Entity Leaves to The Storage

Route Time: 2

Capacity of Storage: 1000

Storage Name: Incoming Storage

Transfer the Product to

○ Route ○ Transporter ● Conveyor

Destination Station: Inspection

Conveyor Name: Inspect Conveyor

Load Time: 0

Figure 3.33 Incoming Docking Module Screen of the Demanufacturing Template

1) Incoming Docking Station Name

A station in a program is a physical location in the model, with a name, for reference in transferring entities. User must fill in the station name or pick up from the drop down list.

2) Arrival Data

This area contains the description of the arrival products which usually come in batch. Mean quantity of the products is required to indicate how many products they have for one arrival. Time between trucks is the time interval of two batches of products which arrival to the facility sequentially. Percentage of products gives the proportion of different products in a batch.

3) Entity Leaves to the Storage

Usually the products are shipped by trucks or other vehicles, then unloaded to the ground. Due to quantity of each batch, the facility needs a temporary storage area to keep a pile of products. The user input is required for information on Route Time, Capacity of Storage, and Storage Name.

4) Transfer the Product to

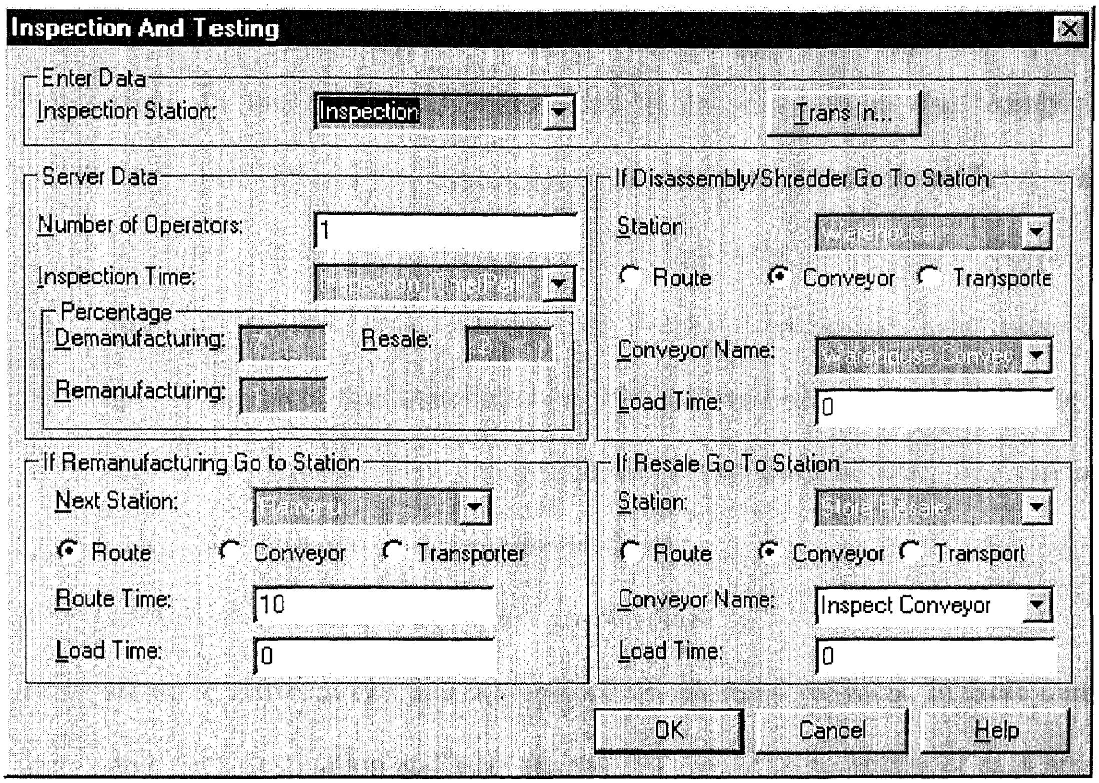
This area determines the transfer mode of batch to the next workstation. There are three options in this frame: Route, Transporter and Conveyor. Each option associate with particular input information. The information requirements of each option are listed in table 3.1.

Table 3.1 Transfer Option Table

Route	Transporter	Conveyor
Destination Station Name	Destination Station Name	Destination Station Name
Route Time	Transporter Name	Conveyor Name
Load Time	Load Time	Load Time
	Capacity	

3.3.3 Inspection and Testing Workstation Module Interface Design

This module is designed to represent the inspection operation in a demanufacturing system. The aim of the Inspection and Testing operation is to detect wear or assess the condition of products. Then, the end fate of the various products is determined. The products found to be in a working condition are sent for resale or to remanufacturing to be repaired and then resold. The end fate of products is classified into 3 stages: Demanufacturing, Remanufacturing, and resale. Figure 3.3 displays the interface design of this module.



Inspection And Testing

Enter Data

Inspection Station: Inspection

Server Data

Number of Operators: 1

Inspection Time: Inspection_Time/Part

Percentage

Demanufacturing: 7 Resale: 2

Remanufacturing: 1

If Disassembly/Shredder Go To Station

Station: Warehouse

○ Route ● Conveyor ○ Transporte

Conveyor Name: Warehouse_Conveyo

Load Time: 0

If Remanufacturing Go to Station

Next Station: Remanu

● Route ○ Conveyor ○ Transporter

Route Time: 10

Load Time: 0

If Resale Go To Station

Station: Store Resale

○ Route ● Conveyor ○ Transport

Conveyor Name: Inspect Conveyor

Load Time: 0

Figure 3.34 Inspection and Testing Module Screen of the Demanufacturing Template

1) Enter Data Area

The information included in this area is the Inspection Station Name, and the Trans In option. When users click "Trans In" button, three options for controlling the transfer device of the entities are displayed on the screen: Transporter, Conveyor, and None, which indicate the transferred mode of a product. Select one option causes the releasing of the particular transfer resource when products arrive to the workstation. This dialog also gives option of including *Unload Time* to account for such activity.

2) Server Data

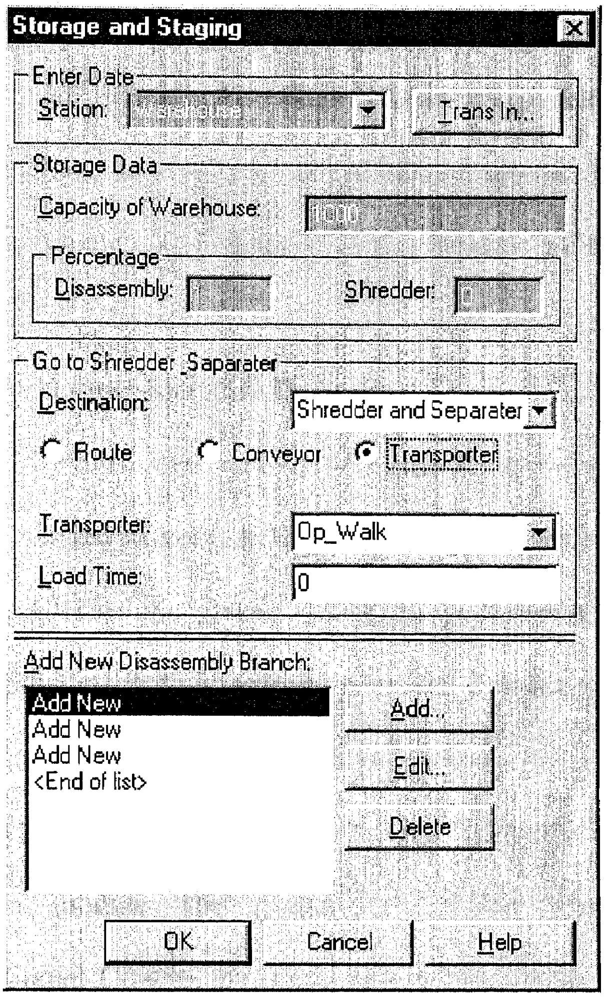
Server data area describes what happens to entities at Inspection and Testing Workstation. In this area, what users need to do is to fill in the Number of Operators which affects the capability of workstation, Inspection Time which performs time delay of inspection operation, and the Percentage for Demanufacturing, Remanufacturing, and Resale. Percentage here means the proportion of products that transferred to their corresponding places. Because the probability of a particular item being detected is not deterministic, one can only approximate the likelihood of serviceable products.

3) If Disassembly/Remanufacturing/Resale Go to Station

These are three different end fate destinations of inspected products. In these frames, users name each destination and push the transfer device radio button of each area to select transfer mode and seize the corresponding transfer resource.

3.3.4 Storage and Staging Module Interface Design

Warehouse is the area which stores all the products waiting for their process in facility. Storage and Staging Module is designed to represent the storage area/warehouse of a facility. The products after inspection that are rejected by remanufacturing or reuse proceed to warehouse waiting for the disassembly or shredder. Screen design is shown in Figure 3.6. The module is classified into four areas: Enter Data, Storage Data, Go to Shredder&Separator and Add New Disassembly Branch.



Storage and Staging

Enter Date

Station: Warehouse

Storage Data

Capacity of Warehouse: 1000

Percentage

Disassembly:

Shredder: 0

Go to Shredder _Saparater

Destination: Shredder and Separater

○ Route ○ Conveyor ● Transporter

Transporter: Op_Walk

Load Time: 0

Add New Disassembly Branch:

Add New
Add New
Add New
<End of list>

Figure 3.35 Storage and Staging Module Screen of the Demanufacturing Template

1) Storage Data

In this area, user specifies the capacity of warehouse, and estimates the proportion of products proceeding to Disassembly Workstation and Shredder and Separator.

2) Go to Shredder and Separator

This area defines the transfer mode to Shredder and Separator Workstation of products. In this area, users need to input the information about the Destination/Shredder Workstation Name, and the options for transfer device. The information relating to each radio button is described in Table 3.1.

3) Add New Disassembly Branch

When users click "Add" button in Add New Disassembly Branch area, a new line is added to the branch window. Additional exit handler appears on the Module Icon. The detailed information about the function of this handler is introduced next.

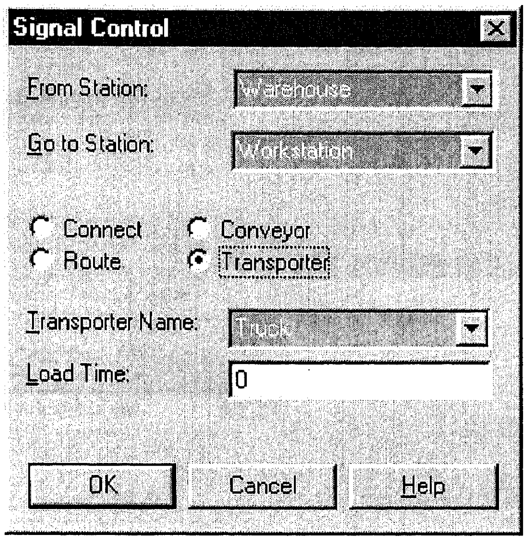
3.3.5 Signal Control Module Interface Design

The disassembly plan determines the number of Disassembly Workstations required for various products in a Demanufacturing facility. From the programming point of view, the programmer needs to allocate the given number of workstations in the screen. Each workstation requests products from warehouse when the number of entities in the queue of disassembly workstation reaches zero. Once warehouse receives the request, it must specify which workstation is asking, and where to send the product. That's the cause we design Signal Control module. This module is a bridge between warehouse and workstations. Every Signal Control module in a program refers to a workstation and keeps the request and delivers the information that is associated with the particular workstation. Meanwhile, each Signal Control module connects with one handler of storage and staging module. Thus, the warehouse can keep track of the request source, and deliver products to the specific destination. Figure 3.7 represents the screen design of this module.

The information required in this module is: The transfer mode of entities, and

From Station: The source station of transfer that is usually warehouse.

Go to Station: The destination station of transfer, that should be the particular workstation.



Signal Control

From Station: Warehouse

Go to Station: Workstation

○ Connect ○ Conveyor
○ Route ● Transporter

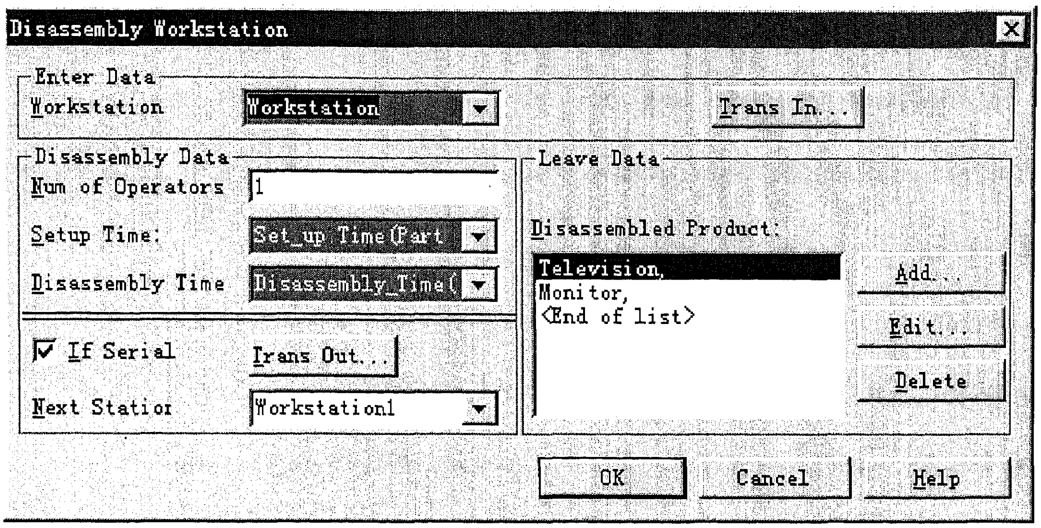
Transporter Name: Truck

Load Time: 0

Figure 3.36 Signal Control Module Screen of the Demanufacturing Template

3.3.6 Disassembly Workstation Module Interface Design

Disassembly is the recovery of materials and subassemblies by separating components in a reverse assembly order. Disassembly Workstation is a production resource for demanufacturing systems. Users are required to input such information as: Disassembly Station Name, Trans-in options, Number of Operators, Setup time, and Disassembly Time as we described in Simulate Module. The module is classified into three areas: Enter Data, Disassembly Data, Leave Data. Disassembly Workstation Screen is shown in Figure 3.8.



Disassembly Workstation

Enter Data

Workstation: Workstation

Disassembly Data

Num of Operators: 1

Setup Time: Set_up Time(Part

Disassembly Time: Disassembly_Time(

☑ If Serial

Next Statio: Workstation1

Leave Data

Disassembled Product:

Television,
Monitor,
<End of list>

Figure 3.37 Disassembly Workstation Module Screen

1) Disassembly Data

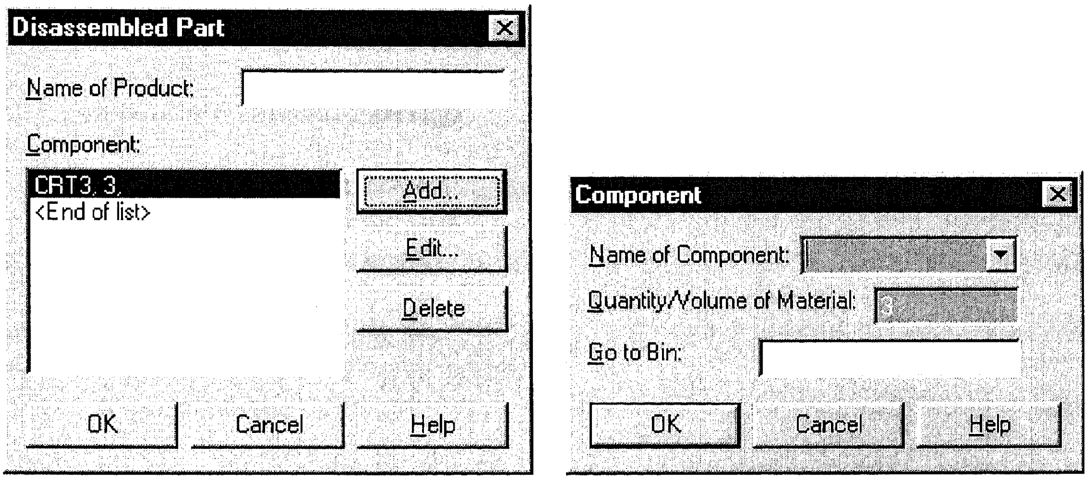
There may be a number of workstations in the system dealing with specific products/subassemblies particularly. The signal control module we described before makes parallel operation of disassembly workstation possible. The "If Serial" option in this module turns out the serial working capability of the workstations. When users select the "If Serial" check box, the "Trans Out" button and Next Station text box appears beside the check box. Then when products/subassemblies are transferred to disassembly workstation. The entities which will be disassembled in current station is unloaded, and the remain entities are then delivered to the next station which specified in the Next Window combo box.

2) Leave Data

In this Leave Data area, we can define all the products that current workstation deal with by filling in the entity name or just picking it up from the entity list.

3.3.7 Disassembled Part Module Interface Design

This Disassembled Part Module represents the part that is disassembled from product/subassemblies with no further parts or subassemblies. The information required in this module relates to the name of component, and quantity/volume of material. When one click "Add" button in Disassembled Part module, Component window pops up and requires the description of a component. Once an additional component is added to this screen, a new handler appears on the user view window of the module which is used to connect with the next module. Figure 3.38 shows the screen design of Disassembled Part Module Screen.

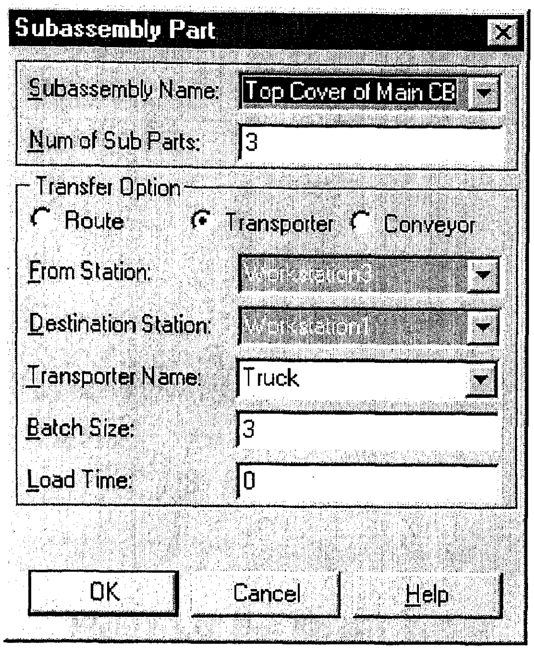


CRT3, 3,
<End of list>

Figure 3.38 Disassembled Part Module Screen

3.3.8 Subassembled Part Module Interface Design

This module focuses on a subassembled part that is disassembled from a product or from other subassemblies. A subassembled part needs to be transferred to the next disassembly workstation and further disassembled to get raw material or component. Information required for this module includes: Subassembly Name, Number of Subassembled Parts, and the Route, Transporter, and Conveyor option for transfer and their corresponding information as described in Table 3.1.



Subassembly Name: Top Cover of Main CB

Num of Sub Parts: 3

Transfer Option

○ Route ● Transporter ○ Conveyor

From Station: Workstation3

Destination Station: Workstation1

Transporter Name: Truck

Batch Size: 3

Load Time: 0

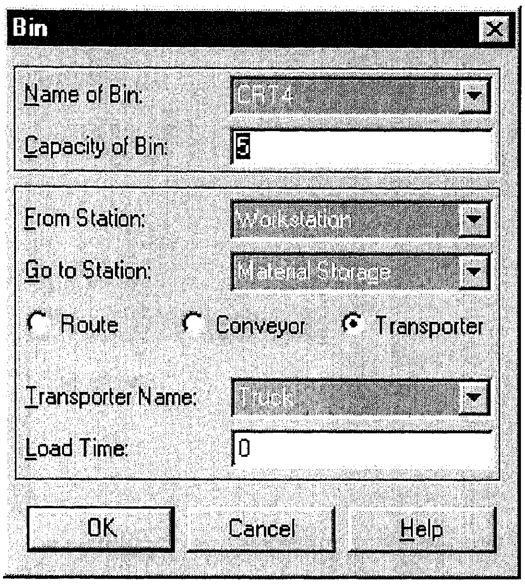
Figure 3.39 Subassembled Part Module Screen

3.3.9 Bin Module Interface Design

A bin is a container used for collecting the disassembled parts and subassembled parts. The disassembly plan determines the number of bins for various recovered components or materials after disassembling the product. The multiple handler of the disassembled part module provides the capability for connection of modules. The user input is required for information on Name of Bin, Capacity of Bin, From Station, Goto Station, and as usual, the transfer option.

Bin module represents a container but not a workstation in the system. Thus, we need to specify the source and destination station of transfer. From station stands for the source station of the transfer, and Go to Station represents the destination of transfer.

Figure 3.11 shows the Bin module Screen.



Bin

Name of Bin: CRT4

Capacity of Bin: 5

From Station: Workstation

Go to Station: Material Storage

○ Route ○ Conveyor ● Transporter

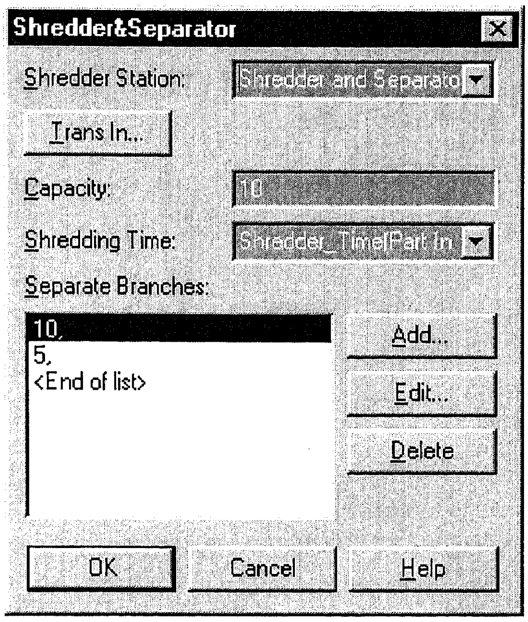
Transporter Name: Truck

Load Time: 0

Figure 3.40 Bin Module Screen

3.3.10 Shredder and Separator Module Interface Design

Shredder and Separator operation is used for volume reduction, and scrap preprocessing. Some products with poor condition observed during inspection and testing or materials from disassembly operations with no life-cycle value are sent to the shredder. The Shredder and Separator module interface in Figure 3.12 is designed in Demanufacturing Template, from which users can input the specific information on Shredder Station Name, the Capacity of Shredder, Shredding Time defined in the Simulate module, and select Trans-In option. During the shredder process, materials like aluminum, steel, and plastic are separated. The Separate Branch box is used to define the separated material such as the Material Name, and Volume/Quantity of the material. When new item is added to the Shredder Branch, an additional exit handler appears on the Shredder and Separator Module. The handler is used to connect with Bins. The materials obtained from Shredder are stored in the Bins first, and the filled bins are to be transferred to Shipping&Staging module.



Shredder Station: Shredder and Separato

Capacity: 10

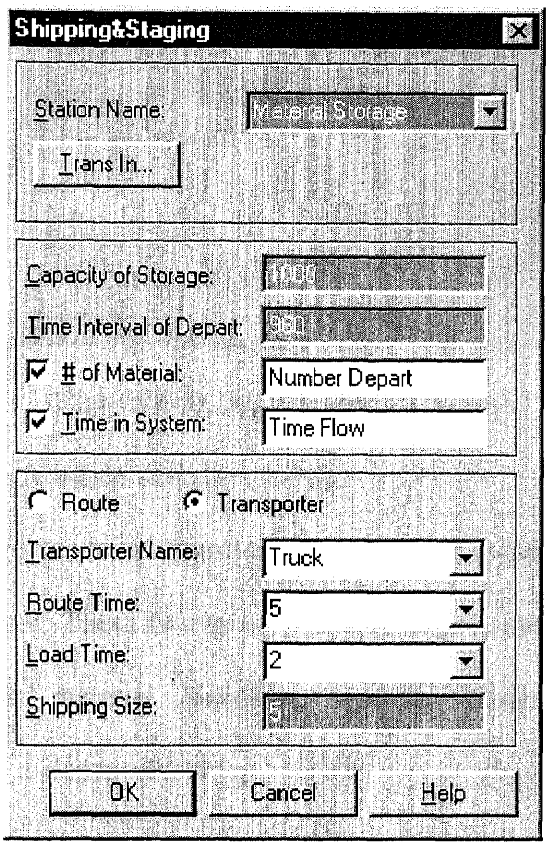
Shredding Time: Shredder_Time(Part In

Separate Branches:
10,
5,
<End of list>

Figure 3.41 Shredder&Separator Module Screen

3.3.11 Shipping&Staging Module Interface Design

Shipping&Staging module can be considered as another warehouse containing materials and components after processing in the system. In another word, all the entities in this storage area are waiting for the resale or disposal. The interface design of Shipping&Staging modules is displayed in Figure 3.13. Users provide Station Name, Trans-in option, Capacity of Storage, Time Interval of Depart, and Trans-out option information as inputs. Time Interval of Depart is the time delay between two departures. Because the time delay of a departure is based on the market and customers, we can only approximate the likelihood of the interval. Therefore, user could fill in a specific time of interval or input distribution expression. The two check boxes: Number of Materials, and Time in System affect the statistic output of simulation. The selection of these two check boxes causes additional items in the performance report. The text boxes beside check boxes are the title of items in the report.



Shipping&Staging

Station Name: Material Storage

Capacity of Storage: 1000

Time Interval of Depart: 960

☑ # of Material: Number Depart

☑ Time in System: Time Flow

○ Route ● Transporter

Transporter Name: Truck

Route Time: 5

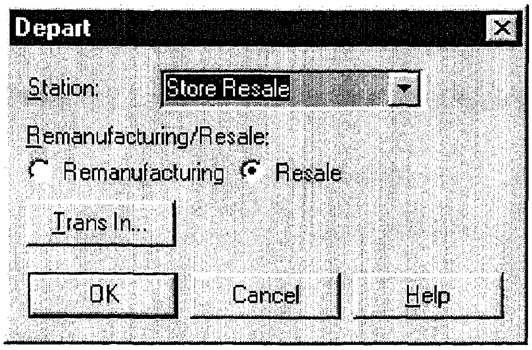
Load Time: 2

Shipping Size: 5

Figure 3.42 Shipping&Staging Module Screen

3.3.12 Depart Module Interface Design

The take-back value and potential reusability of a product are detected and calculated in the Inspection workstation. The end fate of a product is also determined. As we introduced in the previous sections, there are three trans-out directions in Inspection module: Disassembly, Remanufacturing, and Resale. Other than a disassembly station, the specified proportion of products is transferred to Remanufacturing or Resale station. Since our simulation tool aims to simulate demanufacturing facility, remanufacturing and resale operations are not included in the system. Thus Depart module represents as a threshold to remanufacturing/resale workstation, or an exit of the demanufacturing facility. Users are required to provide information on station name and tans-in option, and specify the exit to which workstation.



Depart

Station: Store Resale

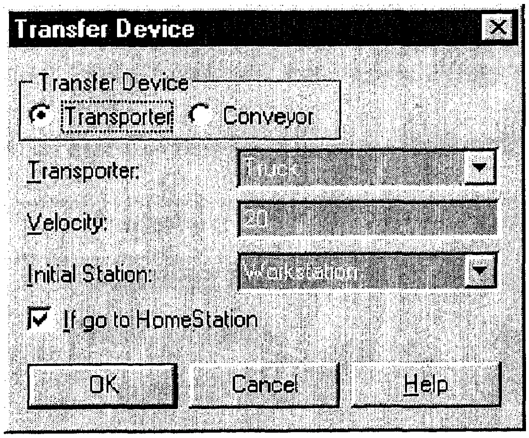
Remanufacturing/Resale:

○ Remanufacturing ● Resale

Figure 3.43 Depart Module Screen

3.3.13 Transfer Device Module Interface Design

Typically, there are two kinds of transfer devices in a demanufacturing system, i.e., Transporter and Conveyor. These two structures in a simulation tool allow us to model almost any type of material handling systems easily. To represent the transporter/conveyor in a demanufacturing system, we should start by first defining a device. The Transfer Device module developed in the simulation template is for the definition capability of a transfer device. The screen design of this module is presented in Figure 3.44.



Transfer Device

Transfer Device

● Transporter ○ Conveyor

Transporter: Truck

Velocity: 20

Initial Station: Workstation

☑ If go to HomeStation

Figure 3.44 Transfer Device Module Screen

Table 3.2 lists the associated input information with different options of Transfer Devices.

Table 3.2 Input Information for Devices

Transporter	Conveyor
Transporter Name Velocity Initial Station Go to Home Station	Conveyor Name Velocity

At the very beginning of the program running, the transporter is allocated at the Initial Station defined by users. If the Go to Home Station is selected, the transporter is forced to go back to initial station/home station when a transfer mission is accomplished.

3.3.14 Distance Definition Module Interface Design

Distance definition module is considered as a connection between each pair of stations or locations which forms roads that entities and transfer devices follow through the facility. Required information for this module is listed in Table 3.3. Each module placed in the program has dual directions. Table 3.4 shows the entries for the distance module.

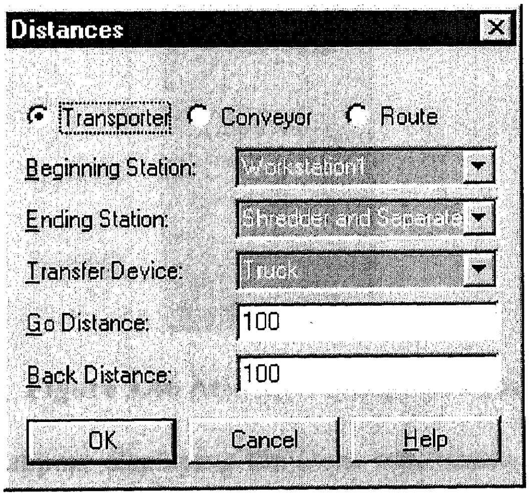
Table 3.3 Screen Entries for Distance Definition module

Route	Transporter	Conveyor
Beginning Station Ending Station	Beginning Station Ending Station Transporter Name Go Distance Back Distance	Beginning Station Ending Station Conveyor Name Segment Length

Table 3.4 The Distance Dialog

Transfer Type Transporter	*Select*
Beginning Station	Workstation1
Ending Station	Workstation2
Transporter Name	Truck
Go Distance	100
Back Distance	100

Once Distance module is placed in the program, two animation paths appear on the screen, Workstation1 → Workstation2 and Workstation2 → Workstation1. If the two directions are not the same path, users can also modify the distance between two stations, and relocate the animation path on the screen. Screen design of this module is shown in Figure 3.16



Distances

● Transporter ○ Conveyor ○ Route

Beginning Station: Workstation1

Ending Station: Shredder and Separate

Transfer Device: Truck

Go Distance: 100

Back Distance: 100

Figure 3.45 Distance Definition Module Screen

3.4 Software Application

Steps to perform System simulation on a demanufacturing facility by MERC System modeling and simulation tool for a demanufacturing facility are specified in this section.

Step1: Start Arena software by double clicking the Arena icon (or shortcut to it).

Step2: Open a new model window by File/New (or new file button on tool bar).

Step3: Attach the panel(s) required to build model. In demanufacturing model, we only need the demanufacturing panel, which you can attach by clicking on the Attach panel tab in the Templates toolbar. This opens the Attach Template Panel dialog in Figure 3.17, where you open demanufacturing panel (file name with extension *.tpo*).

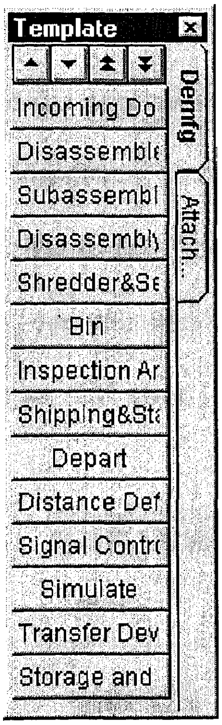


Figure 3.46 Attached Template Panel

The attached panel will appear displaying the icons representing each of the modules in this panel as shown at the left. The buttons in this form of module display are 14 kinds of modules. To add a module to screen, click on its icon in the panel window, move mouse to the location in the model window where you want to position the chosen module, and then click to "drop" it there. This places an instance of the module in the workspace. The placed module always consists of at least a module handle, the box that contains module name and the animation feature or a default name. The animation features normally appear above the module handle and default names normally appear below. When users add their own data to a module, they can change both the animation features as well as the name. Add data to a module by double clicking in its handle to "open it up". This causes the dialog for that module to open as the screen we described before.

From step1-3, the basic simulation environment for a demanufacturing facility is established. The next steps are to build a simulation model to demonstrate how to utilize the simulation tool. Just to illustrate the application, we simplify the demanufacturing facility to a system that has only one inspection, and a disassembly workstations.

Step 4: Put the required modules on the screen: one incoming docking, one inspection, one disassembly workstation, one disassembled part, one bin, one shipping and staging, and one simulate. Figure 3.47 shows the screen for this sample.

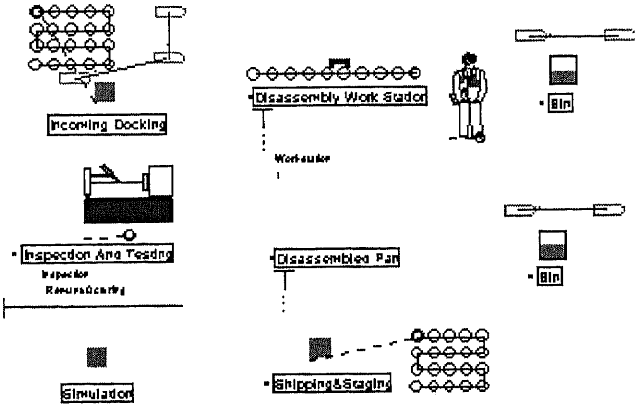


Figure 3.47 Model Window of Placed Modules

Step5: Open 'simulate' module, simulation information: Number of replication and Length of replication are required. The information for product simulated in the system like product name, disassembly time, inspection time, set up time and pictures name for the products are also needed to be entered. The screen is shown in Figure 3.1-3.

Step6: In incoming and staging module in Figure 3.4, the information about Incoming docking Station Name area, Arrival Data area and Entity leave to the Storage area is required to input. Because there have not Transfer Device module in the program, Route option in 'Transfer the Product to' area is selected, and the corresponding information needs to be determined.

Step7: Information in 'Enter Data area', 'Server Data area', 'If Disassembly/Shredder Go to Station' in Inspection Workstation as shown in Figure 3.5 is required to input. 'Depart' module is not utilized in this system, which means there is no product passing to Remanufacturing and Resale stage. Then we don't need to fill out 'If Remanufacturing Go to Station' and 'If Resale Go to station' area, and the proportion in percentage area of 'remanuacturing' and 'resale' should be zero.

Step8: Disassembly workstation name, Set up tools time, disassembly time, operators in disassembly workstation, and the product to be disassembled in this workstation are required to input to Disassembly workstation module as shown in Figure 3.8.

Step9: The disassembled parts are defined in disassembled part module as shown in Figure 3.9. Information about Component and Quantity/Volume of Material is required to input.

Step10: As a container of components, the capacity and name of the bin are required. The transfer information about the bin is also needed to input in the screen presented in Figure 3.11. Connect Disassembled part module and Bin module by using the Connect function from Module menu.

Step11: Station name, Capacity of storage, Time interval of depart, and transfer information are required in this module shown in Figure 3.13.

Step12: The last step in building this model is to arrange the facility layout by relocating the animation feature of the modules and the route path that the entities will follow as they are transferred from one station location to another and the animation feature of the modules. The final module should look something like Figure 3.48.

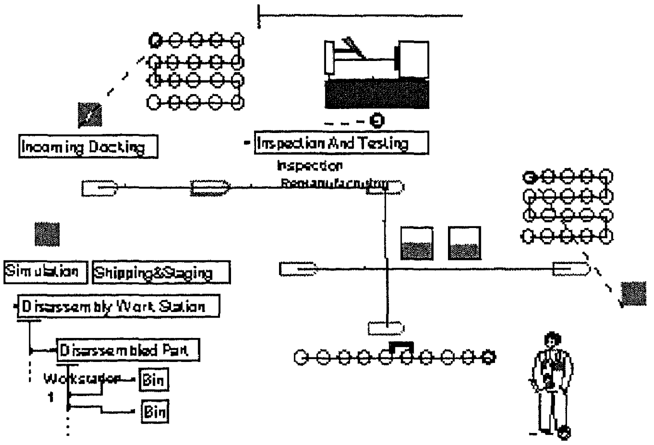


Figure 3.48 The Final Model

Step13: Now we are ready to run the simulation. Normally there are three ways to run the simulation: Use the Go button on the run toolbar, the Run/Go option, or the F5 key.

Step14: if users run the simulation to completion, a dialog pops up and asks if users want to see the result. If uesrs select yes, they should get a window of the summary report of the simulation.

CHAPTER 4

CASE STUDY ON DEMANUFACTURING FACILITY

4.1 Introduction

In this chapter, we select a typical small demanufacturing facility as demanufacturing simulation tool validation examples. The facility under study is a typical demanufacturing firm. The objective of the facility is the disassembly of electronics equipment with intent to ship the disassembled parts to outside vendors for further processing. Initially, the information about the desired facility is collected and analyzed. Second, logic design is considered by selecting proper modules and putting them into the new project screen. Then information description about facility and simulation is entered to each module on screen. Finally, the layout of the facility required in project is arranged. As a result, the operational report of the system is presented. The prime objective of template development and system simulation is to evaluate an actual or planned system, and identify opportunities for operational improvements of that system. A small electronics demanufacturing facility is used here as a specific example to show how to utilize system simulation with demanufacturing template, evaluate and identify a system in the following sections.

The system to be modeled consists of part arrivals, inspection cells, a warehouse, three disassembly cells, a shredder cell, and part departures. Each workstation has a single operator. Electronic products incoming into the facility are televisions, computers, phone, and vacuums. Material handling equipment includes one fork truck and two conveyors. Out of total products 25% are monitor, 5% are vacuums, 20% are television, and 50% are printer. The data collection sheets and analysis for the processing time and the products arriving information are presented in Appendix A. The data on products arriving into facility was collected for a period of three months starting from May to July 1998 for the report sheet in the facility. The data collected for vacuum cleaner and televisions was under observation for two different days and eight hours per day [Ketan 18].

4.2 Software Application

4.2.1 Modeling Development

To perform a simulation on demanufacturing facility, we first need to open Arena software by double clicking the Arena icon in the Windows program manager. The program starts up. Then we need to attach the demanufacturing template created by MERC and open a new model window file to create the simulation program.

- **Defining the Data**

The data modules we need to place for this model are shown in Figure 6.1. We have already covered almost all of the needed concepts contained in these modules in the previous chapters. We only comment on each and discuss any issues of which we should be aware.

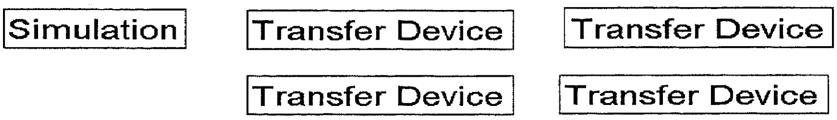


Figure 4.1 The Data Modules

Users need to enter the simulation and product processing information into the simulation module, and material handling information into the transfer device modules. We have placed a total of 4 transfer device modules (two for fork lift and two for conveyor). To place modules into the screen, move the mouse to the module and click once to select it, then move mouse to model window and click to place the module. If users double click the handle of that module, user screen will pop up to let users input the corresponding information. After users input the available data, they should click "OK" button to save the change to that module. Tables 6.1 to 6.5 display the information for the data modules.

Table 4.1 Completed Simulation Module

Project	
Project	Demanufacturing
Analyst	MERC
Date	4/20/2000
Number of Replication	5
Length of Replication	4800
Product Definition	
Product	Television
Product Index	1
Inspection Time	TRIA(0.17, 0.348, 1.95)
Disassembly Time	10.5 + GAMM(2.81, 2.16)
Setup Time	TRIA(0.06, 0.315, 0.54)
Product	Monitor
Product Index	2
Inspection Time	0.63 + WEIB(1.02, 1.85)
Disassembly Time	NORM(15.2, 1.84)
Setup Time	TRIA(0.06, 0.315, 0.54)
Product	Vacuum
Product Index	3
Inspection Time	0.16 + 1.84 * BETA(1.41, 1.32)
Disassembly Time	8 + ERLA(0.791, 5)
Setup Time	TRIA(0.06, 0.315, 0.54)
Product	Printer
Product Index	4
Inspection Time	1 + ERLA(0.48, 2)
Disassembly Time	15 + 21 * BETA(0.539, 1.17)
Setup Time	TRIA(0.06, 0.315, 0.54)
Picture Definition	
Picture Name	
Television	Television
Monitor	Monitor
Vacuum	Vacuum
Printer	Printer

Table 4.2 Completed Transfer Device Module for Forklift1

Transfer Device	
Transporter	*Select*
Transporter Name	Forklift1
Velocity	20
Initial Station	Workstation 1
If go to Home Station	*Check*

Table 4.3 Completed Transfer Device Module for Forklift2

Transfer Device	
Transporter	*Select*
Transporter Name	Forklift1
Velocity	20
Initial Station	Workstation 1
If go to Home Station	*Check*

Table 4.4 Completed Transfer Device Module for Warehouse Conveyor

Transfer Device	
Conveyor	*Select*
Conveyor Name	Warehouse Conveyor
Velocity	10

This module defines the conveyor between inspection workstation and Storage staging area, which transfers the products passing the inspection to warehouse.

Table 4.5 Completed Transfer Device Module for Inspect Conveyor

Transfer Device	
Conveyor	*Select*
Conveyor Name	Inspect Conveyor
Velocity	5

This module defines the conveyor between incoming storage and inspection workstation. The Inspect conveyor transfers product from the incoming storage area to inspection workstation.

- **Create Arrival Product and Direct to Warehouse**

  After their arrival, based on the condition of products under inspection, some of them are transferred to remanufacturing and resale area, and the others are passed to warehouse waiting for the disassembly. The logic design of these functions is presented in Figure 4.2.

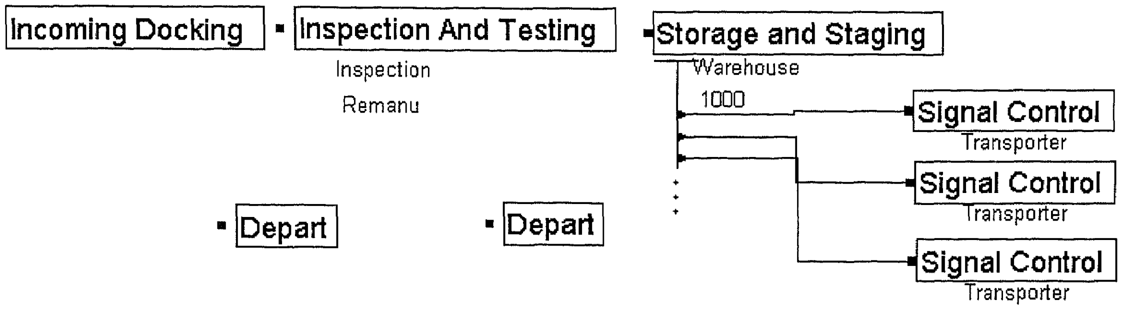


Figure 4.2 Arrival and Service Selection Process Logic

We use Incoming Docking to create the products appearing in the system. Inspection and testing operation makes three branches for the product transfer direction. Two Depart modules stand for Remanufacturing and Resale area respectively, and Storage and Staging module holds the products for the disassembly operation. Tables 4.6-14 display the information of the modules.

Table 4.6 Completed Incoming Docking Module

Station Name	Arrival Station
Mean Quantity of Product	EXPO(20)
Time Between Arrivals	TRIA(20,30,31)
Percentage of Products	
Monitor	0.25
Television	0.3
Vacuum	0.5
Printer	1.0
Transfer to Incoming Storage	
Route Time	2
Capacity of Storage	1000
Storage Name	Incoming Storage
Transfer Product to	
Conveyor	*Select*
Destination	Inspection
Conveyor Name	Inspect Conveyor

Table 4.7 Completed Inspection Module

Inspection Station	Inspection
Number of Operator	1
Inspection Time	Inspection_Time
Percentage	
Demanufacturing	0.7
Resale	0.2
Remanufacturing	0.1
Remanufacturing	
Next Station	Remanufacturing
Route	*Select*
Route Time	10
Resale	
Next Station	Resale
Route	*Select*
Route Time	10
Demanufacturing	
Next Station	Warehouse
Conveyor	*Select*
Conveyor Name	Warehouse Conveyor

Table 4.8 Completed Depart Module for Remanufacturing Area

Remanufacturing	*Select*
Station	Remanufacturing
Trans-In	
None	*Select*

Table 4.9 Completed Depart Module for Resale Area

Resale	*Select*
Station	Resale
Trans-In	
None	*Select*

Table 4.10 Completed Storage and Staging Module for Resale Area

Station	Warehouse
Trans-In	
Conveyor	*Select*
Capacity of Warehouse	1000
Percentage	
Disassembly	1.0
Shredder	0
Destination	Shredder
Transporter	*Select*
Transporter Name	Forklift1
Add New Disassembly Branch	3 New Branches

Table 4.11 Completed Signal Module for Workstation1

From Station	Warehouse
Destination Station	Workstation1
Transporter	Select
Transporter Name	Forklift1

The Signal Control module represented in Table 4.11 makes the connection between Warehouse and Workstation1, which distinguishes the product request form the workstation1 and sends command to transfer products to a particular workstation. The other two Signal modules are almost the same as the presented one, the changes for the other two modules are listed in Figure 4.12-4.13.

Table 4.12 Change of Signal Module for Workstation1

Destination Station	Workstation2

Table 4.13 Change of Signal Module for Workstation1

Destination Station	Workstation3

- **The Disassembly Modules**

The responsibility of each workstation is an essential issue to manipulate the system. Workstations in this model deal with different products and disassembly workstations in the facility are working in a parallel batch operation method. Figure 4.3 shows the logic for disassembly workstation design. The detailed information is shown in Tables 4.14 to 4.24.

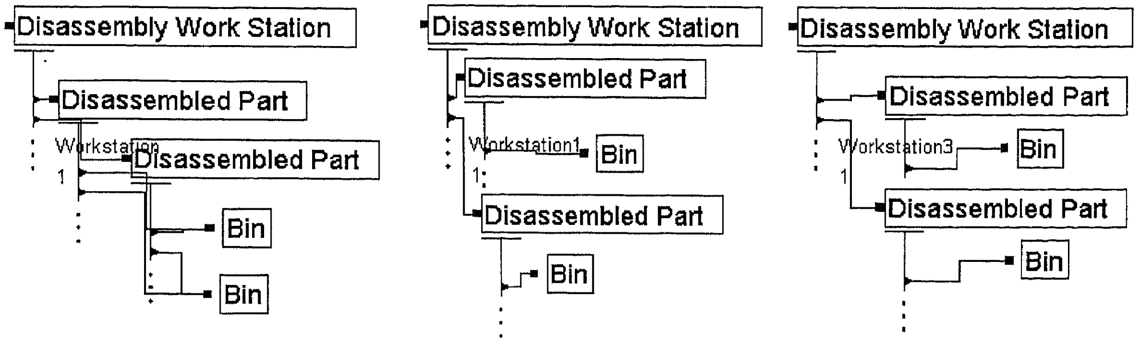


Figure 4.3 Disassembly Modules

We use the disassembly workstation to perform the disassembly operation and each kind of products being disassembled in the workstation is associated with a Disassembled part module. The disassembled materials or components are defined in the Disassembled part module. After disassembly, the material is placed in the bins, which is presented by the Bin module. Each bin contains the particular components obtained from a disassembly operation. A filled bin is transferred to a specified destination.

Table 4.14 Completed Disassembly Workstation Module for Workstation1

Workstation Name	Workstation1
Trans-In	
Transporter	Select
Number of operators	1
Setup time	Setup_Time
Disassembly Time	Disassembly_Time
Disassembled Product	Monitor
	Television

The changes for workstation2 and workstations from workstation on the workstation name and disassembled product are listed in Tables 4.15-16.

Table 4.15 Changes of Disassembly Workstation Module for Workstation2

Workstation Name	Workstation2
Disassembled product	Vacuum

Table 4.16 Changes of Disassembly Workstation Module for Workstation2

Workstation Name	Workstation3
Disassembled product	Printer

Table 4.17 Completed Disassembled Part Module Three Workstations

Disassembled Part Module 1 for Workstation1	
CRT	1
Disassembled Part Module 2 for Workstation1	
Circuit Board	3
Disassembled Part Module 3 for Workstation2	
Wires	30
Disassembled Part Module 4 for Workstation2	
Steel	3
Disassembled Part Module 5 for Workstation3	
Plastic	3
Disassembled Part Module 6 for Workstation3	
Aluminum	2
Dumped Material	3

Each product should have several components or materials disassembled through the operation. But to simplify the simulation program and clarify the report from the simulation, we just pick up some typical samples from a variety of components to indicate the processing of a disassembly operation.

Table 4.18 Completed Bin Module for CRT

Name of Bin	CRT
Capacity of Bin	10
From Station	Workstation1
Go to Station	Shipping Storage
Transporter	Select
Transporter Name	Forklift1
Load Time	3

The different entries for the other Bins are listed in the Table 4.19-4.24.

Table 4.19 Bin Module for Circuit Board

Name of Bin	Circuit Board
Capacity of Bin	30
From Station	Workstation1

Table 4.20 Bin Module for Wire

Name of Bin	Wire
Capacity of Bin	500
From Station	Workstation2

Table 4.21 Bin Module for Steel

Name of Bin	Steel
Capacity of Bin	20
From Station	Workstation2

Table 4.22 Bin Module for Plastic

Name of Bin	Plastic
Capacity of Bin	20
From Station	Workstation3

Table 4.23 Bin Module for Aluminum

Name of Bin	Aluminum
Capacity of Bin	20
From Station	Workstation3

Table 4.24 Bin Module for Dumped Material

Name of Bin	Dumped Material
Capacity of Bin	50
From Station	Workstation3
Go to Station	Shredder
Transporter Name	Forklift1

Click “OK” to save the changes users make to the module. After the definition of each module, users need to connect the related modules (Disassembly workstation-Disassembled part, Disassembled part-Bin) by using the “Connect” tool from the toolbar of Arena or specifying the next module via “GO TO” text box in the Disassembly module and Disassembled part module. For instance, users can connect a disassembly workstation module with disassembled part module via “Connect” tool provided by Arena. First click the “Connect” button. Second, move mouse to the exit point on disassembly workstation module and click once. Third move mouse to the entry point of disassembled part module and click once. Then a connection line appears between these two modules. The entities from the disassembly workstation will be transferred to the disassembled part through the connection line.

- **Shredder and Shipment of Products**

Some disassembled products in Bins of recovered materials and components are directly transferred to the Shipping and staging area where they are sold to vendors. Some of them are sent to shredder and then after shredding they are sent to the shipping and staging area. The logic design for shredder and shipping is presented in Figure 4.4.

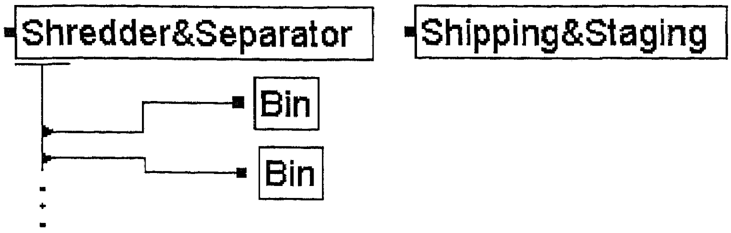


Figure 4.4 Logic Design for Shredder and Shipping Operation

The Shredder and Separator module performs the shredder operation in the system, and the shredded batch of material is stored in Bins. Each bin in the logic is associated with a specific kind of material. Materials processed in the system are sent out by the shipping and staging module. All the information about shredder, shipping modules and the Bins for shredder Material is listed in Tables 4.25-28. The changes for the other bin from the one described in table 4.26 are presented in Table 4.27.

Table 4.25 Completed Shredder and Separator Module

Station Name	Shredder
Trans-In	
Transporter	Select
Capacity	10
Shredding Time	Shredder_Time
Separate Branches	
Volume	10
Volume	5

Table 4.26 Completed Bin Module for Shredded Plastic

Name of Bin	Shr_Plastic
Capacity of Bin	10
From Station	Shredder
Go to Station	Shipping Storage
Transporter	Select
Transporter Name	Forklift1
Load Time	3

Table 4.27 Bin Module for Shredded Metal

Name of Bin	Shr_Metal
Capacity of Bin	15

Table 4.28 Completed Shipping and Staging Module

Station Name	Shipping
Trans-In	
Transporter	Select
Capacity of Storage	1000
Time Interval of Depart	960
Route	Select
Route Time	20
Shipping Size	50

Click the "OK" button to save the changes you have made to each module and connect the Shredder and Separator module with Bin modules as we described previously.

4.2.2 Facility Layout Arrangement

After the definition of the operational modules, the last step in building this model is to add the route paths that the entities follow from one station location to another. We add these paths using the Distance Definition module from the demanufacturing template. Click on this module in the demanufacturing panel, and place it into the model window. The Distance Definition module is shown as Figure 4. 4.

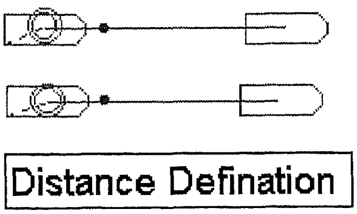


Figure 4.5 User View of Distance Definition Module

Users are required to specify the information needed for the distance definition module which influences the facility layout. The facility layout is formed by areas such as inspection, and disassembly workstations and the routes among them. The distance definition module gives the animation feature of the stations and routes.

We need to place a module of the distance between each pair of stations or locations. In this case, we need to place 21 separate modules. The needed information for a distance module is shown in Table 4.29. The module placed in the model has dual-directions. Distance between warehouse and disassembly workstation1 in the case of Table 4.21, from warehouse to workstation and from workstation to warehouse may be different if the paths are different. The rest of distance modules are similar to this one. The distance between each pair of stations is shown in Figure 4.5

Table 4.29 Completed Distance Module

Transporter	*Select*
Beginning Station	Warehouse
Ending Station	Workstation1
Go Distance	20
Back Distance	20

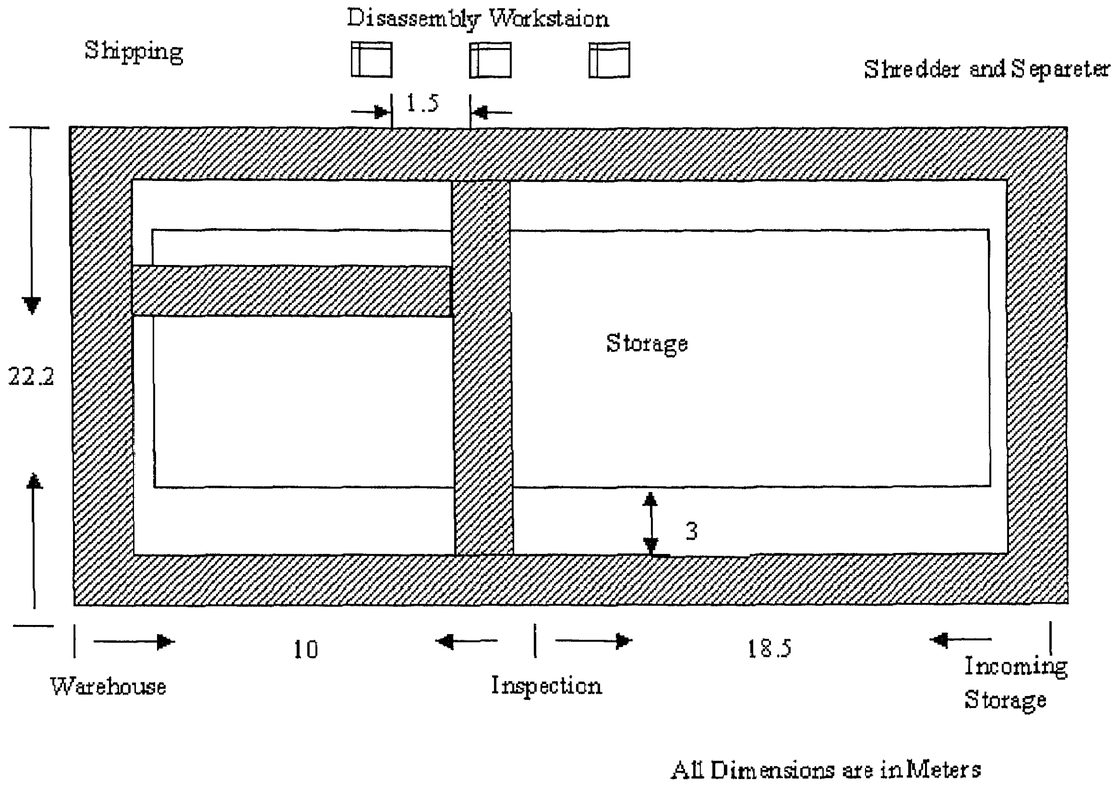


Figure 4.6 Distance Description of Facility Layout

After the Distance Definition module is defined, we need to consider the animation component of a Distance module, which consists of two stations connected by a distance. When a part is transported from one station to another, the transfer device follows this line (or distance) on the animation. To incorporate these animation objects in the model window, we move the stations to their correct locations and position the line so it follows the layout path. Then we can place the Bin objects beside the disassembly workstation or any other animation object related to the stations. The final disassembly layout is similar to the one in Figure 4.6.

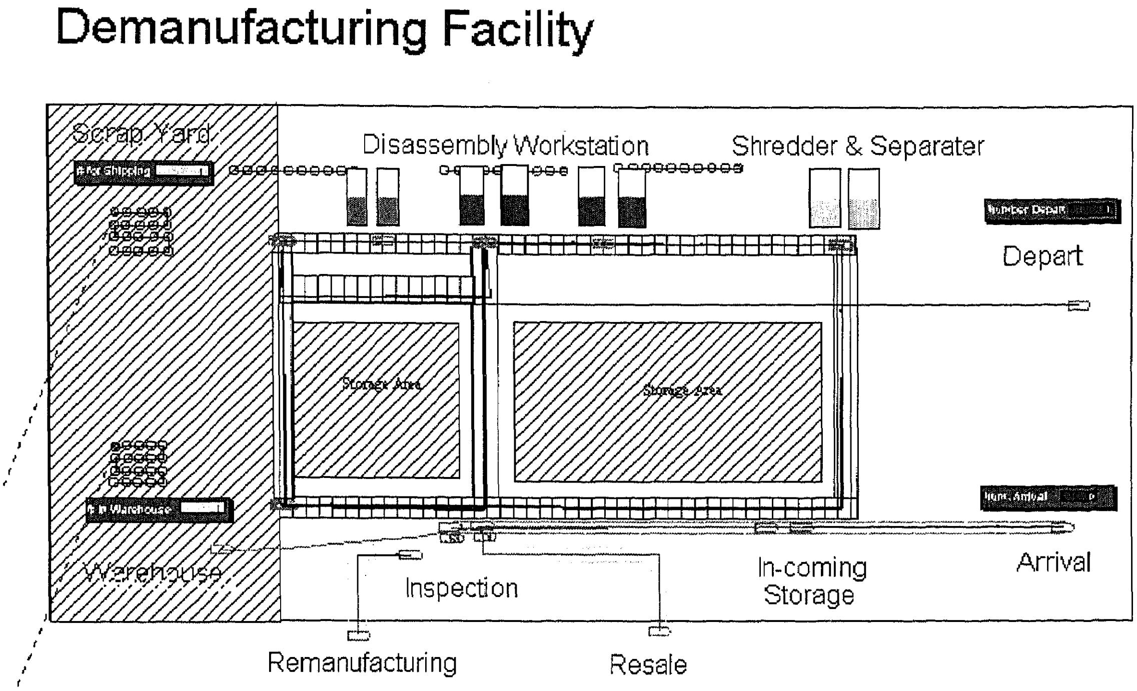


Figure 4.7 Demanufacturing Facility Layout

4.2.3 Results and Analysis

To run the model, click on the "Run" button in the run toolbar, or select "Go" from the Run menu. After the execution, the analysis report is obtained by summarizing information in the simulation. From the animation and the report analysis, users can better understand the facility. Figure 4.7 shows the workstations time allocation statistics for the simulated facility. From the Figure, it can be observed that the workstation3 has a better utilization, about eight-one percent of time is utilized for disassembly operation. Meanwhile, workstation1 and workstation2 have a long idle time, which means the efficiency of these two workstations needs to be improved.

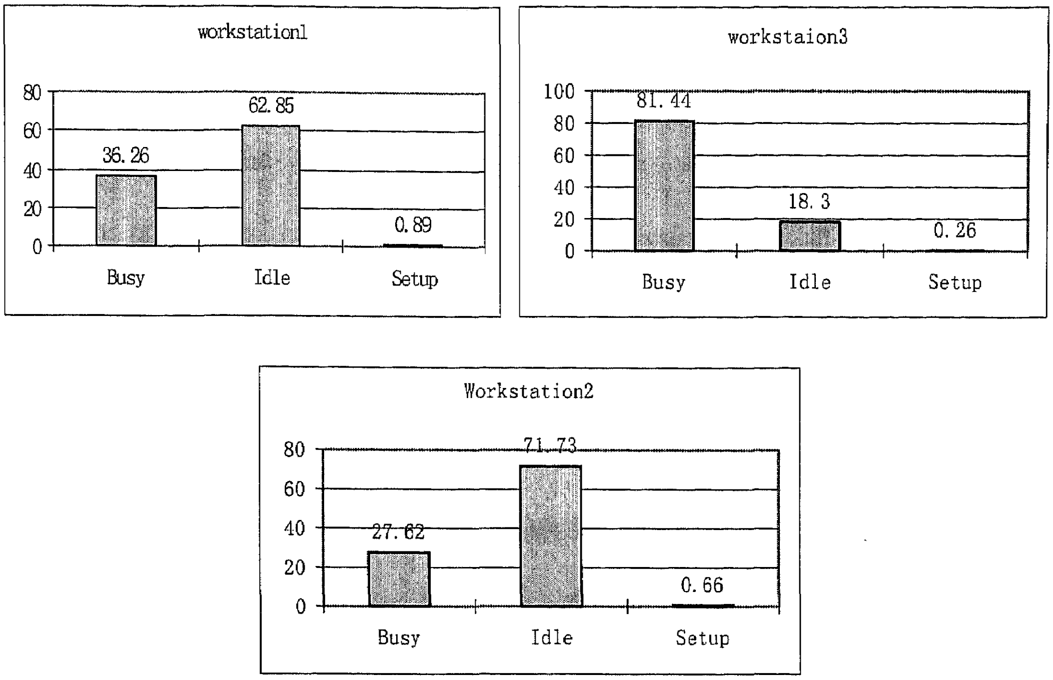


Figure 4.8 Results on Three Workstation Time Allocations

Figure 4.9 shows the Truck Time Allocation. The simulation results show that Truck1 is too busy during the operation, and may cause the unavailability of the service for the workstation which requests transporter. Truck1 is busy during approximately ninety-nine percent of the time as shown in the Figure4.9.

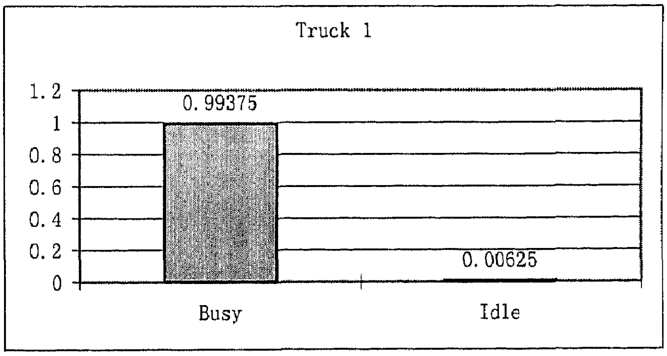


Figure 4.9 Results on Transporter Time Allocations

One problem is that operator idle time is greater than the operator busy time in workstation1 and workatation2. It may be caused by the unavailability of transporter for the workstations. The material handling device allocated for transfer between workstations is over utilized, another device may be required to facility the transport. The below alternative to the current scenario introduces one more truck to resolve this problem.

4.2.4 Alternative Operational Strategy Improvement

The current operations statistics and above observations reveal the inefficiency and bottleneck in existing methods of operation and also suggests improvements in the current operation.

Due to the observation in Section 4.2.2, the main suspect about the inefficient utilization of resource Workstation1 and Workstation2 is lack of the material handling devices. The improvement option employs Truck2 to assist Truck1 for the material handling between workstations and the warehouse. Table 4.22 shows the change made for Signal module.

Table 4.30 Change of Signal Modules

Transporter Name	Truck2

The result of the simulation of the alternative scenario reports a great improvement of the utilization of the workstations. Figure 4.9 shows the time utilization of three Disassembly Workstations.

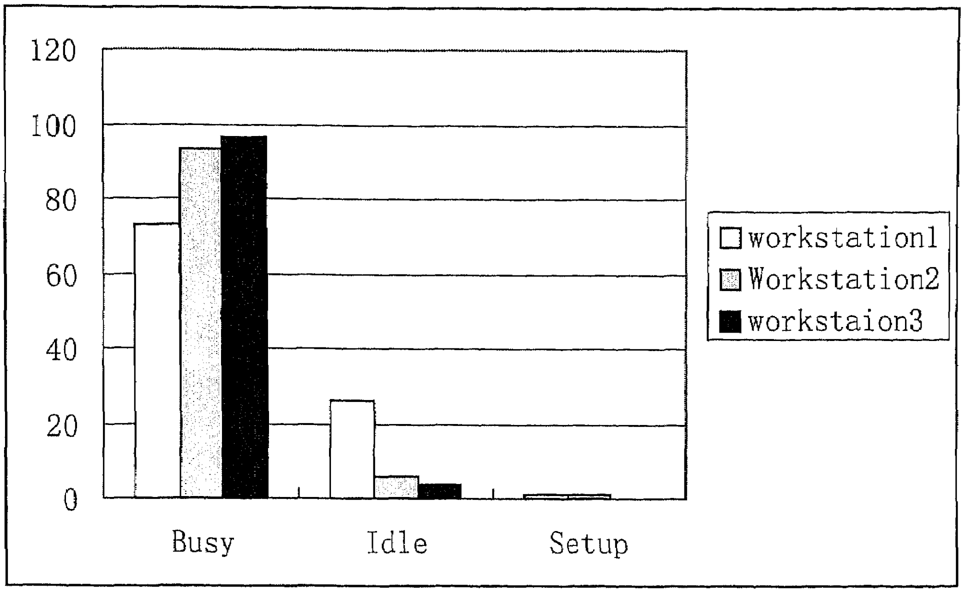
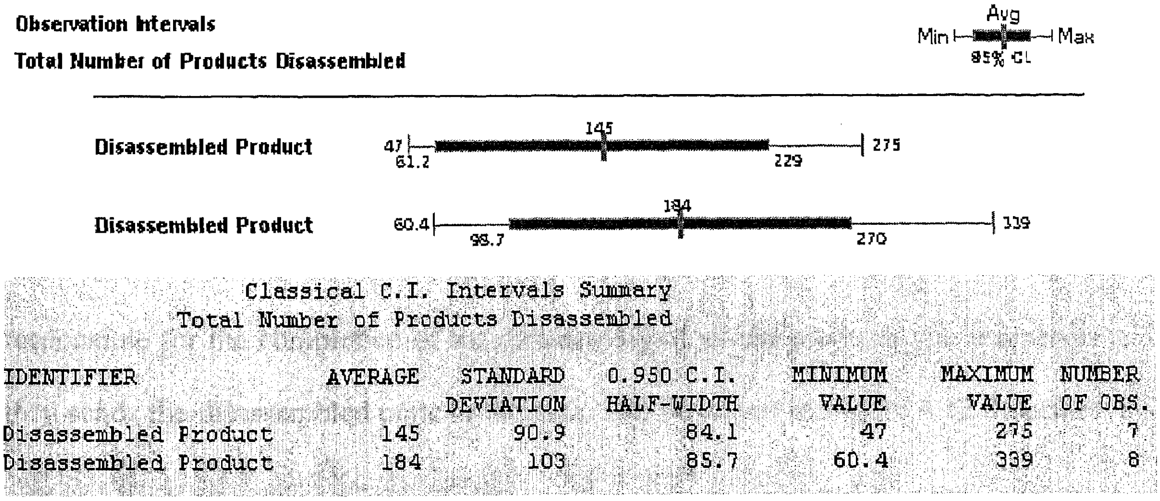


Figure 4.10 Time Utilization of Disassembly Workstation

In comparison with first scenario there is an obvious increase in Busy processing time. It can be observed from Figure 4.7 that the idle time in workstation1 and workstation3 is reduced.

The alternative scenario also leads to an increase in the throughput of the facility. Figure 4.11 shows the variation, at 95 confidence level in total number of products disassembled during the simulation period.



```
                    Classical C.I. Intervals Summary
                 Total Number of Products Disassembled

IDENTIFIER               AVERAGE   STANDARD   0.950 C.I.    MINIMUM     MAXIMUM  NUMBER
                                   DEVIATION  HALF-WIDTH     VALUE       VALUE   OF OBS.
Disassembled Product        145       90.9        84.1         47         275       7
Disassembled Product        184       103         85.7        60.4        339       8
```

Figure 4.11 Total Number of Products Disassembled

The total number of products disassembled in the first scenario is 145. The average number of products disassembled in the second scenario is 184. From this observation, the total number of products disassembled in the alternative scenario is higher than the original one.

With the simulation tool, the "what if" questions and the suggestions to improve the system efficiency provided by users can be easily simulated and analyzed. The report of the simulation for different scenarios finally helps users to make the best decisions.

4.3 Comparison of Different Disassembly Workstation Configurations

Another study on an efficient demanufacturing process includes different facilities with different configuration of the equipment. All the facilities are configured to receive an equal quantity of products. The products that disassembled in the facility are computers. The simulated facility consists of four disassembly workstations. In the first scenario, the workstations work in serial mode. The computer flows through all the workstation. Each one of them handles a part of the disassembly assignment. The first workstation accepts the products from the warehouse, disassembles them and sends the remaining products to the next disassembly workstation. The next workstation accepts the product form the previous workstation, performs the disassembly and ships the remaining products to the next workstation and so on. Finally, the leftover products reach the last disassembly workstation, which performs the disassembly on those products. The last workstation is responsible for the completion of the disassembly of all the products that it receives and then sends the disassembled parts to the bins. The flowchart in Figure 4.12 describes this configuration. Computers stay in each workstation for three minutes to accomplish the process. Table 4.31 shows the inspection, disassembly and setup time for the computer.

Table 4.31 Processing Time of Computer

Product Name	Disassembly Time	Inspection Time	Setup Time
Computer	EXPO(3)	EXPO(1)	1

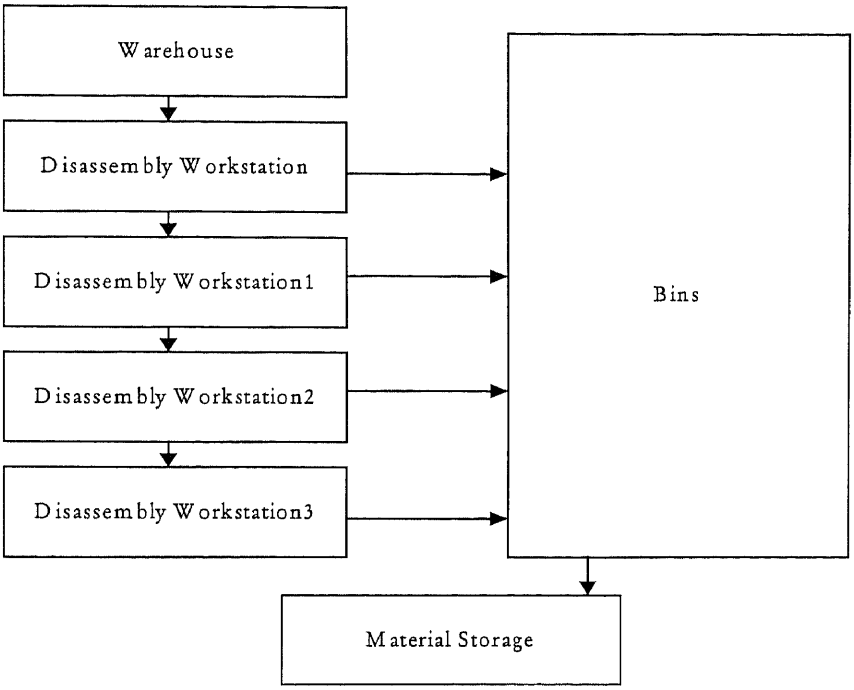


Figure 4.12 Flow Chart for 4 Serial Disassembly Workstations.

Each arrival of the batch includes 200 computers. As the time interval for the arrival batch changes, throughput of the facility changes. The time intervals we used for this facility are 0.5, 1, 2, 3, 4 and 5 days respectively. After execution of the model, the results are shown in Figure 4.13.

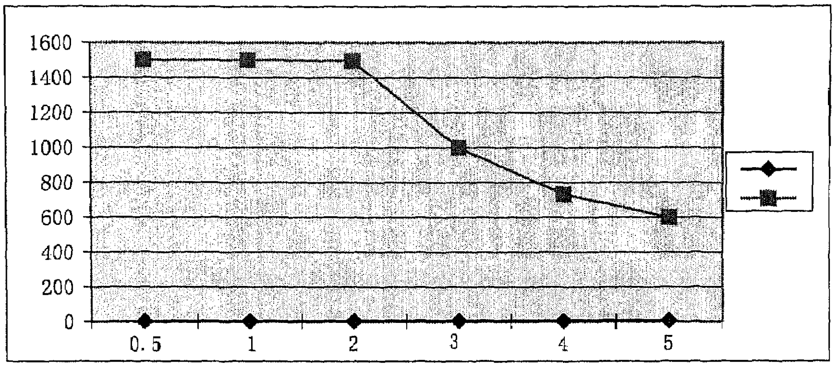


Figure 4.13 The Throughput for Different Time Intervals of Serial Workstations

From Figure 4.13, the larger the interval time, the less the throughput. We also notice that the time intervals of 0.5, 1 and 2 days lead to almost the some throughputs. This means that the product arrival speed has exceeded the material handling speed, and products will be accumulated in the warehouse. It seems that the 2 days interval for the facility is the balance point for the material processing among the tested intervals. The configuration of the facility is shown in Figure 4.14.

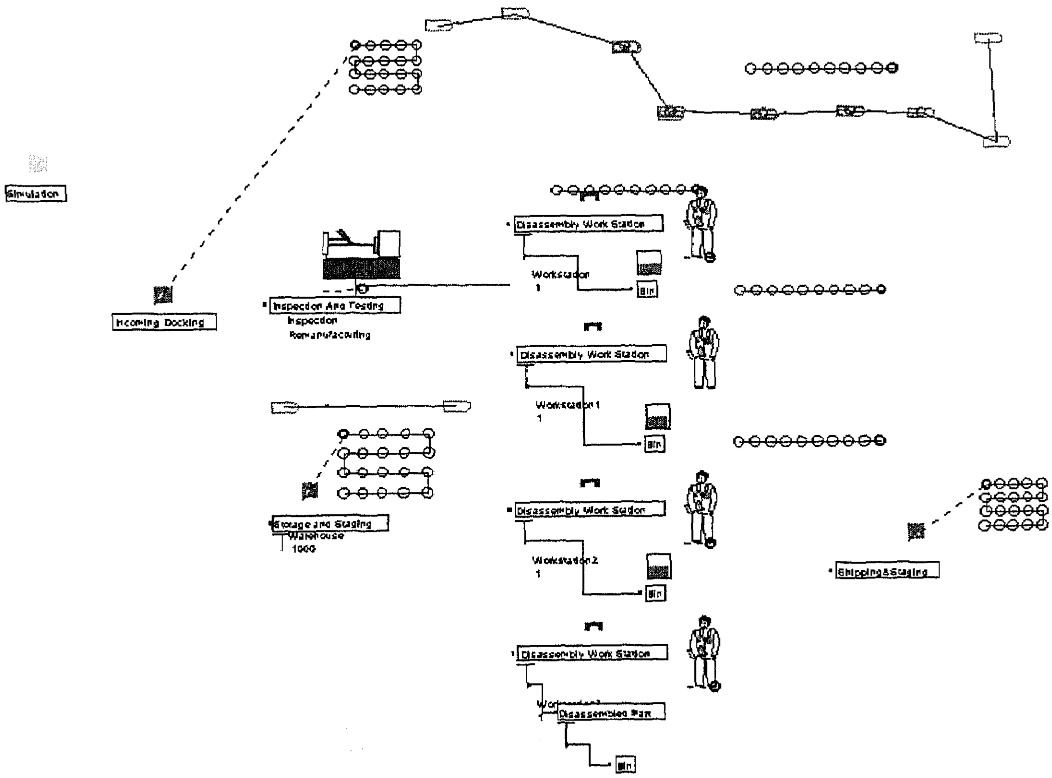


Figure 4.14 Configuration of the Facility of Serial Workstations

In the second scenario, disassembly workstations work in a parallel mode. Every workstation can get products from the warehouse and process the disassembly activities. Because all the disassembly operations are accomplished in one workstation, the disassembly time for each computer in the disassembly workstation change to EXPO(13).

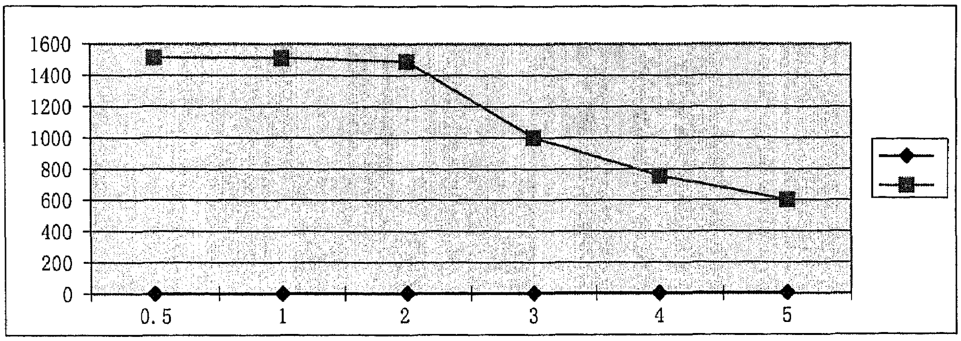


Figure 4.15 The Throughput for Different Time Intervals of Serial Workstations

The configuration of the second scenario with parallel workstations are shown in Figure 4.16.

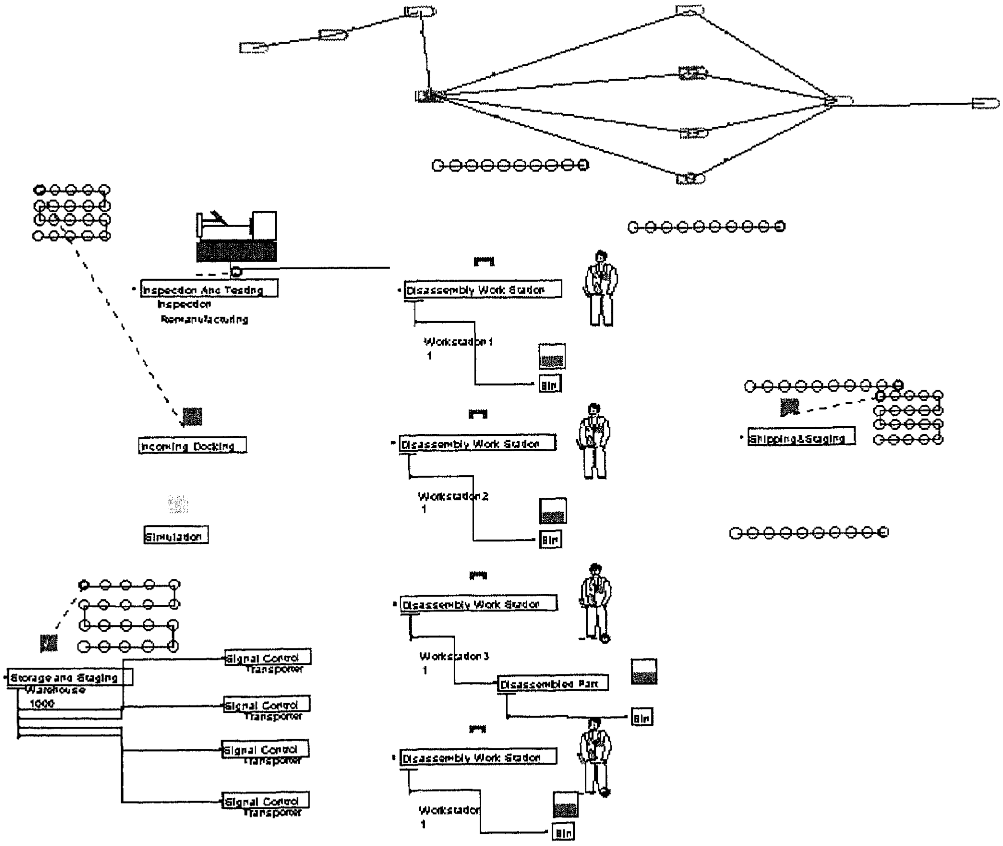
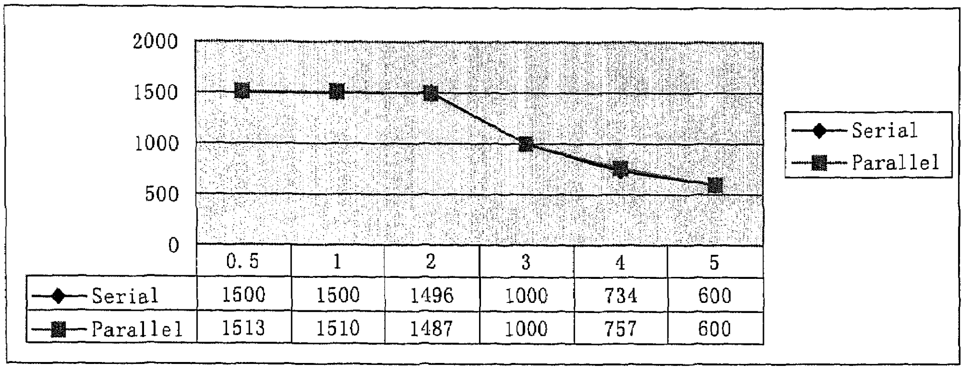


Figure 4.16 Configuration of the Second Scenario with Parallel Workstations

Form the report, we can compare the throughput of these two scenarios. The results are shown in Figure 4.17.



	0.5	1	2	3	4	5
Serial	1500	1500	1496	1000	734	600
Parallel	1513	1510	1487	1000	757	600

Figure 4.17 Comparison of the Throughput for Serial and Parallel Workstations

From Figure 4.17, one can observe that there is no big difference between the throughputs of serial workstations and parallel workstations. The throughputs for parallel workstations are a little bit higher than the serial one at the time interval 0.5 and 1. Note that the transportation time required between serially arranged workstations is ignored. More importantly, the equal disassembly times among them are assumed. A workstation in the parallel configuration may be more expensive since it needs to deal with all the disassembly tasks for a product. Mixed configuration can also be simulated.

CHAPTER 5

CONCLUSION

5.1 Contributions of this Thesis

The traditional lifecycle stage assessment approach is extended in Multi-lifecycle Assessment to consider remanufacturing, remanufacturing, and reengineering stages. As value recovery stage, demanufacturing gains more attention through the lifecycle stages, and it is necessary to consider the system efficiency of a demanufacturing operation. Simulation provides the ability to mimic and analyze the demanufacturing system. The modeling concepts and the simulation development methodology are introduced. Because of the knowledge gap between the demanufacturer and the simulator, a simulation tool for demanufacturing facility is desired. A completed initial prototype version of the demanufacturing simulation tool with fourteen modules in the template is successfully developed.

A demanufacturing simulation tool is designed to simplify the development of a system simulation program for demanufacturing facility. Such a program can be used to evaluate demanufacturing facility layout and operational strategies. The simulation tool is developed by enhanced functionality of a commercially available simulation software Arena. The simulation tool provides capability to evaluate the utilization of resource, evaluate the facility layout, identify bottleneck in a system, evaluate alternative equipment changes, and analyze operational strategies. The differences between the simulation tool and the commercial available simulation software are that the simulation tool bridges the gap between the hard core simulation knowledge and the demanufacturing terminology, make the demanufacturing simulation development easy to understand and use. Based on the common activities of demanufacturing operation, the developed simulation tool embraces the following customized modules for demanufacturing facility:

1. Simulation
2. Incoming Docking
3. Inspection
4. Depart
5. Storage and Staging
6. Signal Control
7. Disassembly Workstation
8. Disassembled Part
9. Subassembled Part
10. Bin
11. Shredder and Separator
12. Shipping and Staging
13. Transfer Device and
14. Distance Definition

The detailed design on demanufacturing simulation tool including the user interface and logic design is provided. The demaufacturing simulation tool is validated based on data collected from a typical demanufacturing facility. Alternative scenario is evaluated to improve the performance of facility. Serial and parallel configurations of disassembly stations are evaluated.

5.2 Limitations and Future Research

The utilization of a simulation modeling tool is more than a modeling approach in analyzing a system which focuses on the system efficiency or resource utilization. This requires a clear vision and understanding of the demanufacturing operations from the product collection to disassembly process. Initial version of a demanufacturing simulation tool is completed, but the software needs to be extended to provide more robust application development.

Some of the recommendations for the demanufacturing simulation tool are listed below:

- Development of special functions and algorithms for generating and evaluating more complex decisions in scheduling and planning to provide more flexibility in machine selection and sequencing operation.
- Improve graphic outputs by using and incorporating Visual Basic application into the software.

APPENDIX

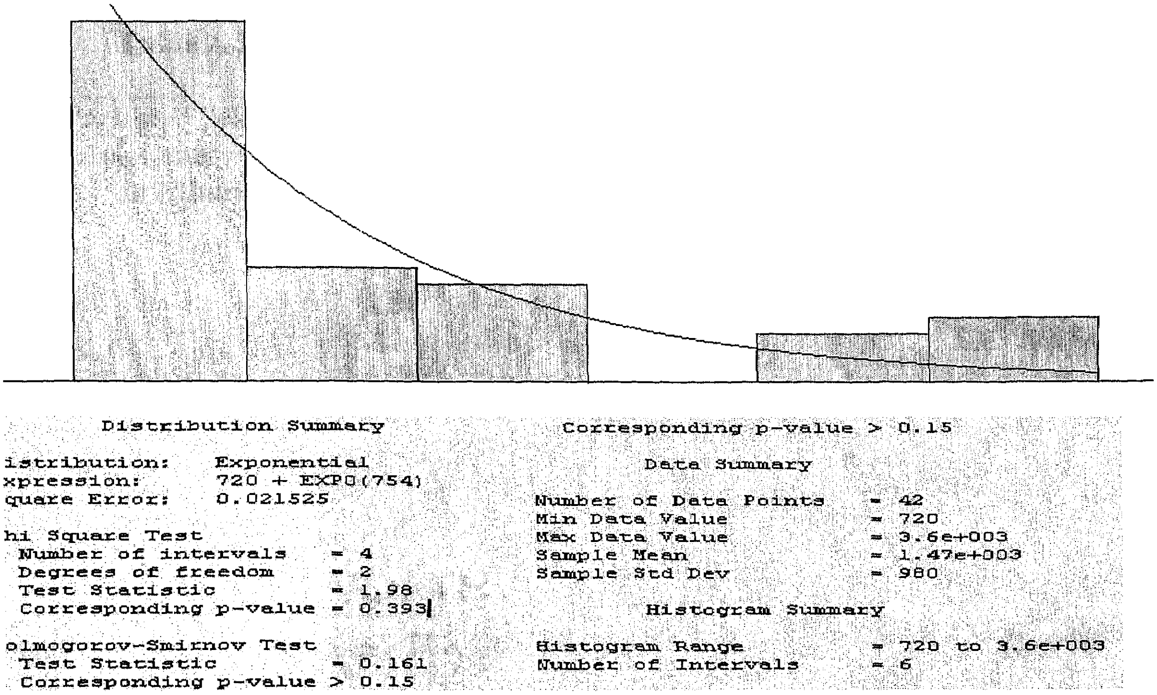
DATA AND DISTRIBUTIONS ON PRODUCT ARRIVALS

Table 1 Data on Product Arrivals [18]

Difference in days	No of pallets	Difference in days	No of pallets
2	1	1	4
1	3	1	4
2	2	1	3
1	8	3	3
3	4	1	30
1	6	2	3
2	6	1	8
5	1	3	1
2	15	1	4
4	12	1	14
1	8	3	4
5	5	2	4
1	42	1	2
1	1	1	3
1	16	3	2
5	32	3	32
1	41	1	2
1	27	4	2
4	11	2	22
1	18	5	5
1	24	1	29
		1	4

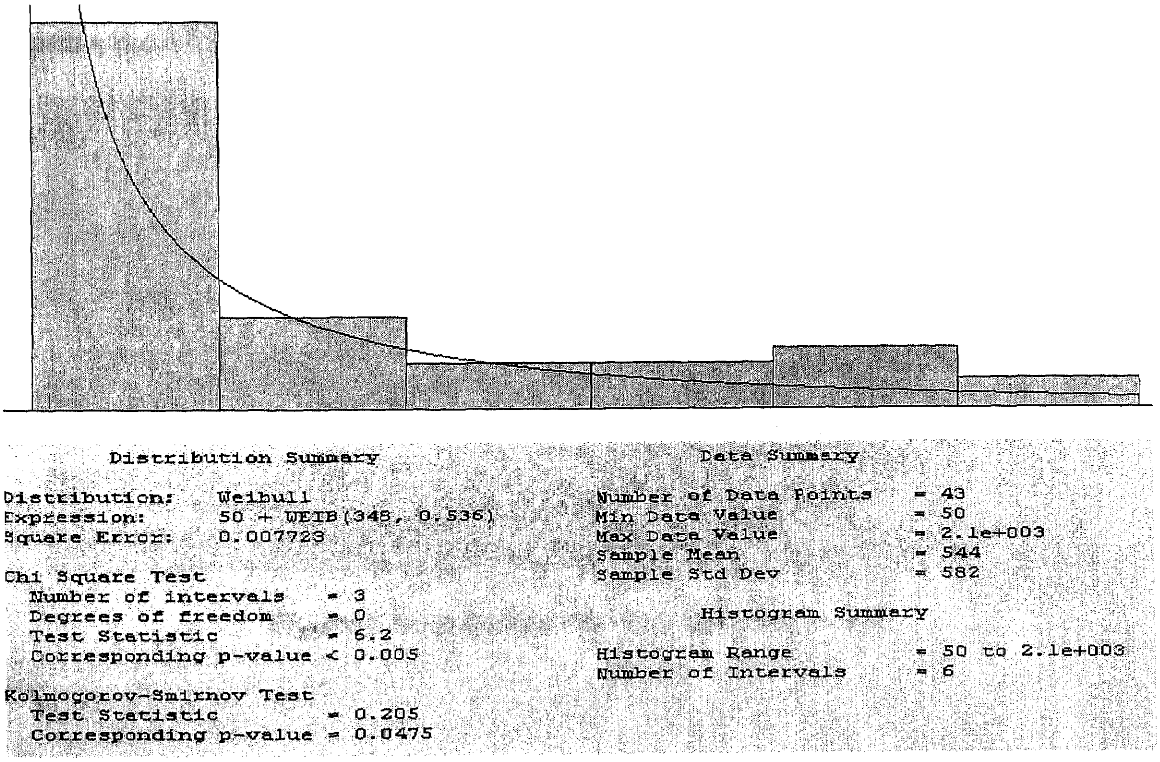
Pallets per month	Average Pallets
May-129 June-113 July-226	156

APPENDIX A (Cont.)



```
        Distribution Summary                    Corresponding p-value > 0.15

istribution:      Exponential                          Data Summary
xpression:        720 + EXPO(754)
quare Error:      0.021525                   Number of Data Points    = 42
                                             Min Data Value           = 720
hi Square Test                               Max Data Value           = 3.6e+003
 Number of intervals    = 4                  Sample Mean              = 1.47e+003
 Degrees of freedom     = 2                  Sample Std Dev           = 980
 Test Statistic         = 1.98
 Corresponding p-value  = 0.393                    Histogram Summary

olmogorov-Smirnov Test                       Histogram Range          = 720 to 3.6e+003
 Test Statistic         = 0.161              Number of Intervals      = 6
 Corresponding p-value  > 0.15
```

Figure 1 Distribution for Inter-arrival Times



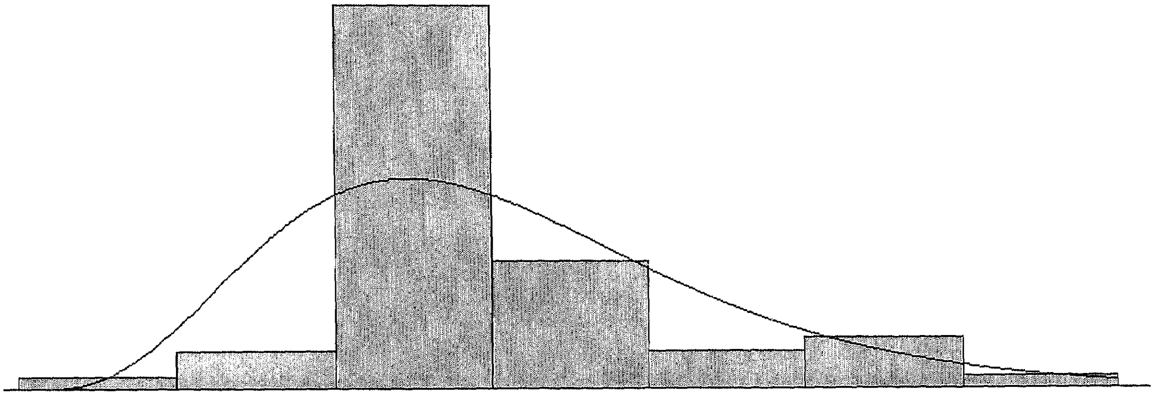
```
        Distribution Summary                          Data Summary

Distribution:     Weibull                    Number of Data Points    = 43
Expression:       50 + WEIB(348, 0.536)      Min Data Value           = 50
Square Error:     0.007723                   Max Data Value           = 2.1e+003
                                             Sample Mean              = 544
Chi Square Test                              Sample Std Dev           = 582
  Number of intervals    = 3
  Degrees of freedom     = 0                       Histogram Summary
  Test Statistic         = 6.2
  Corresponding p-value  < 0.005             Histogram Range          = 50 to 2.1e+003
                                             Number of Intervals      = 6
Kolmogorov-Smirnov Test
  Test Statistic         = 0.205
  Corresponding p-value  = 0.0475
```

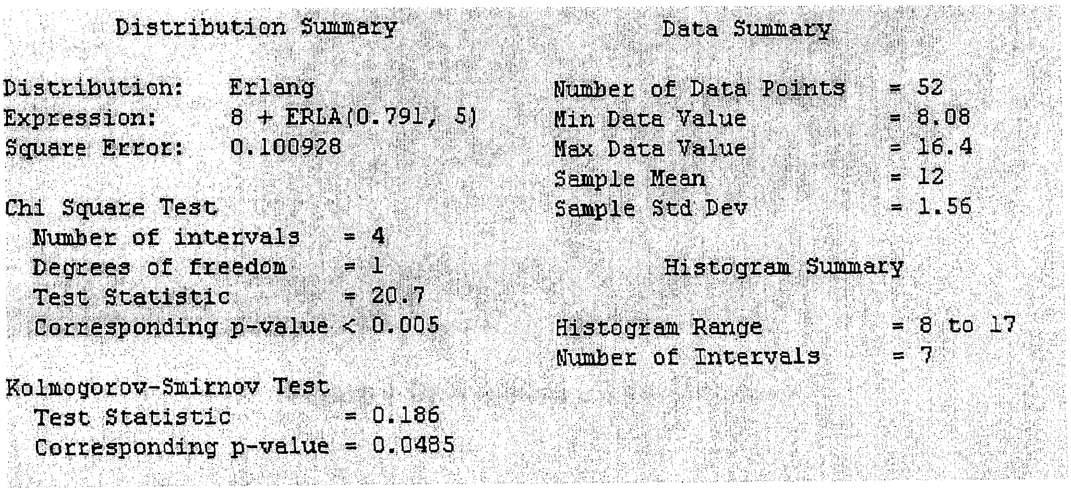
Figure 2 Distribution for Number of Products

APPENDIX (Cont.)

Table A.2 Disassembly Times in Minutes for Vacuum Cleaner [18]

Vacuum Cleaners								
Disassembly Times in Minutes	2.58	1.55	1.19	1.79	3.13	2.58	1.75	1.52
	2.25	1.33	1.1	0.91	1.17	1.36	1.14	0.92
	5.16	1.3	1.59	1.33	1.19	1.4	2.08	1.87
	0.8	1.49	1.38	4.3	1.44	1.47	1.52	1.98
	1.36	1.4	1.11	1.66	1.63	1.7	1.63	1.84
	2.24	1.51	1.33	1.66	1.83	1.36	1.83	1.87
	1.78	1.26	2.16					





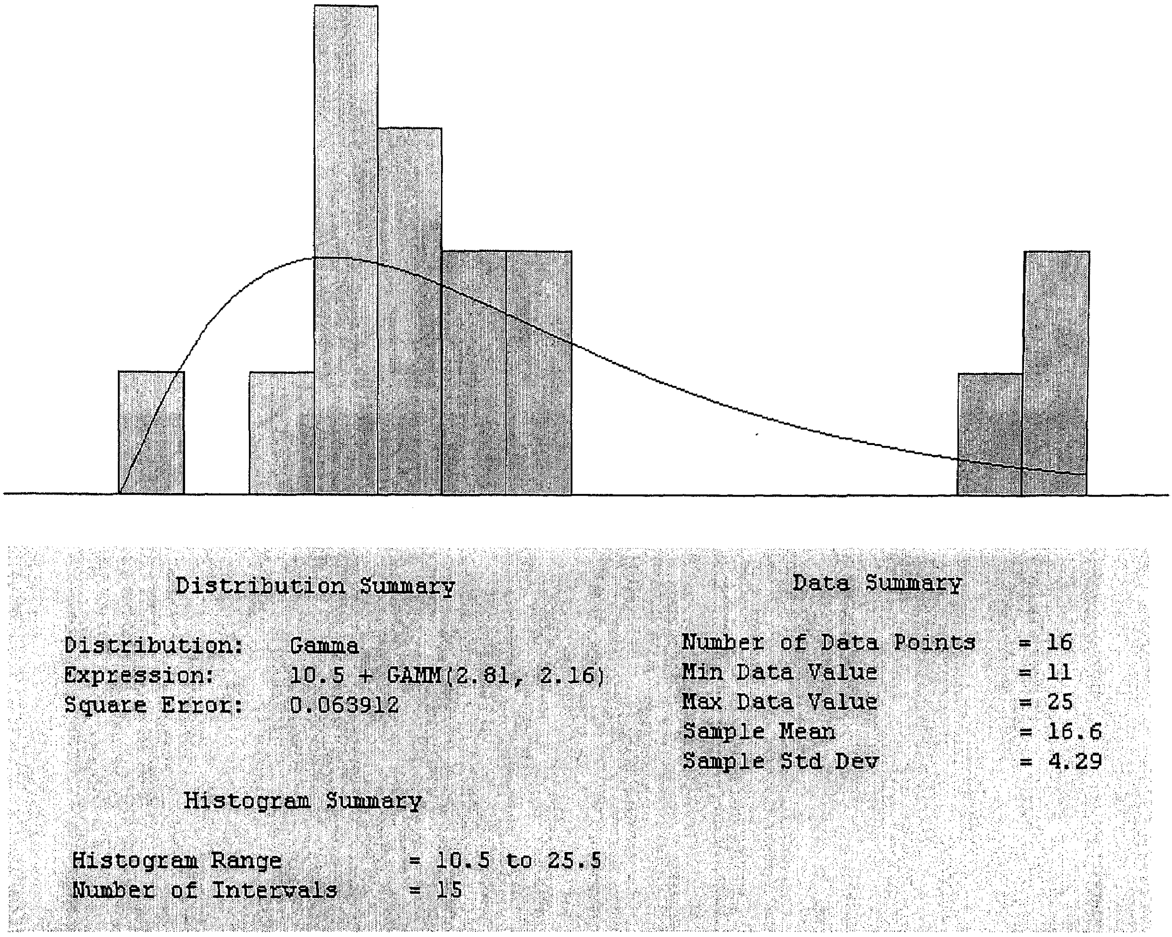
```
        Distribution Summary                          Data Summary

Distribution:    Erlang                   Number of Data Points    = 52
Expression:      8 + ERLA(0.791, 5)       Min Data Value           = 8.08
Square Error:    0.100928                 Max Data Value           = 16.4
                                          Sample Mean              = 12
Chi Square Test                           Sample Std Dev           = 1.56
  Number of intervals     = 4
  Degrees of freedom      = 1                    Histogram Summary
  Test Statistic          = 20.7
  Corresponding p-value < 0.005           Histogram Range          = 8 to 17
                                          Number of Intervals      = 7
Kolmogorov-Smirnov Test
  Test Statistic          = 0.186
  Corresponding p-value = 0.0485
```

Figure 3 Distribution Fit for Vacuum Cleaner

APPENDIX (Cont.)

Table 3 Disassembly Times in Minutes for Television [18]

Televisions								
Disassembly Times in Minutes	5 25	6 4	6 5	5 3	4 11	7 4	4 7	4 15



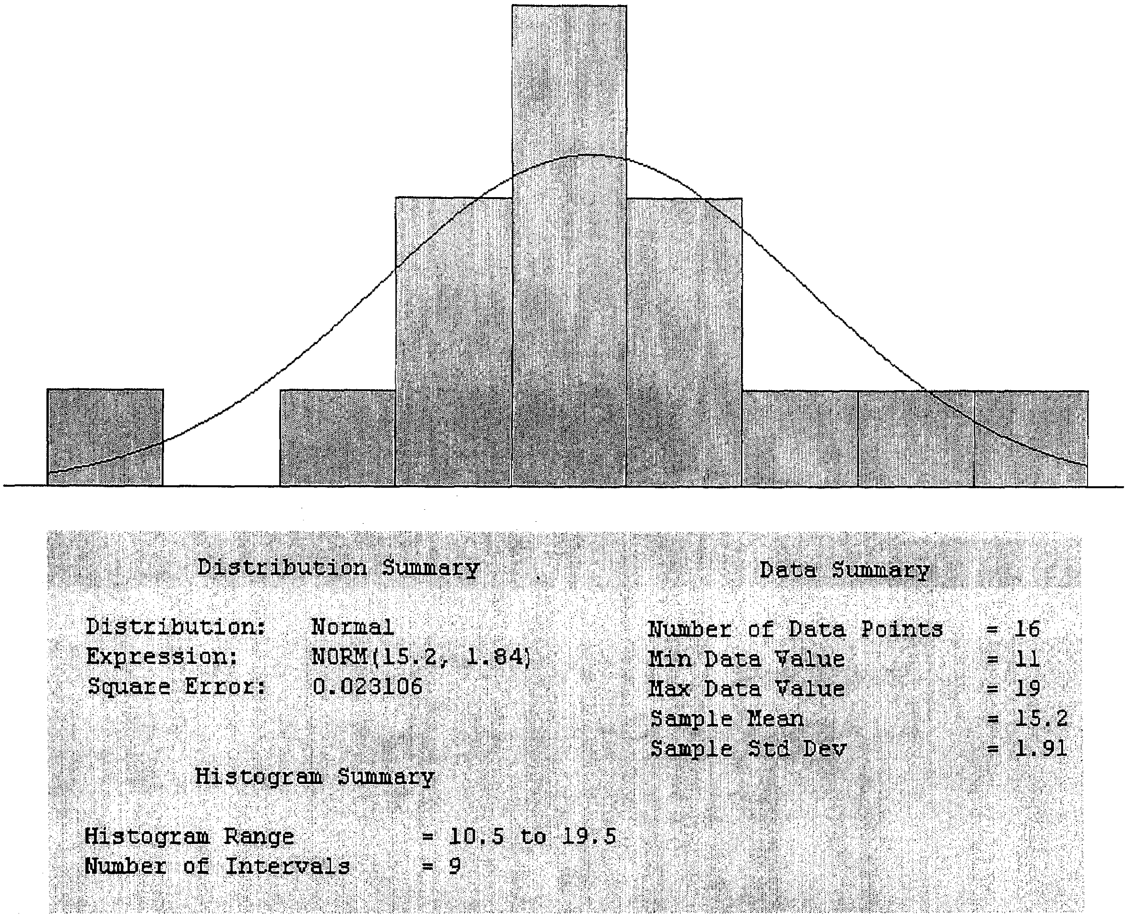
```
        Distribution Summary                              Data Summary

Distribution:    Gamma                          Number of Data Points   = 16
Expression:      10.5 + GAMM(2.81, 2.16)        Min Data Value          = 11
Square Error:    0.063912                       Max Data Value          = 25
                                                Sample Mean             = 16.6
                                                Sample Std Dev          = 4.29
          Histogram Summary

 Histogram Range          = 10.5 to 25.5
 Number of Intervals      = 15
```

Figure 4 Distribution Fit for Television

APPENDIX (Cont.)

Table 4 Disassembly Times in Minutes for Monitor

Televisions								
Disassembly Times in Minutes	15	16	14	17	15	18	11	16
	19	15	15	13	14	16	14	15



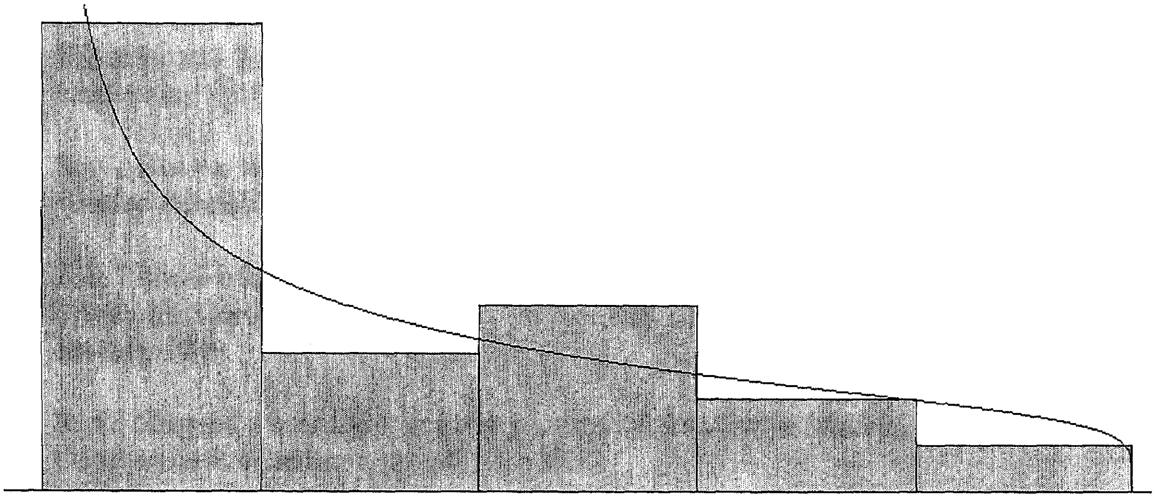
```
        Distribution Summary                          Data Summary

Distribution:    Normal                 Number of Data Points   = 16
Expression:      NORM(15.2, 1.84)       Min Data Value          = 11
Square Error:    0.023106               Max Data Value          = 19
                                        Sample Mean             = 15.2
                                        Sample Std Dev          = 1.91
        Histogram Summary

Histogram Range         = 10.5 to 19.5
Number of Intervals     = 9
```

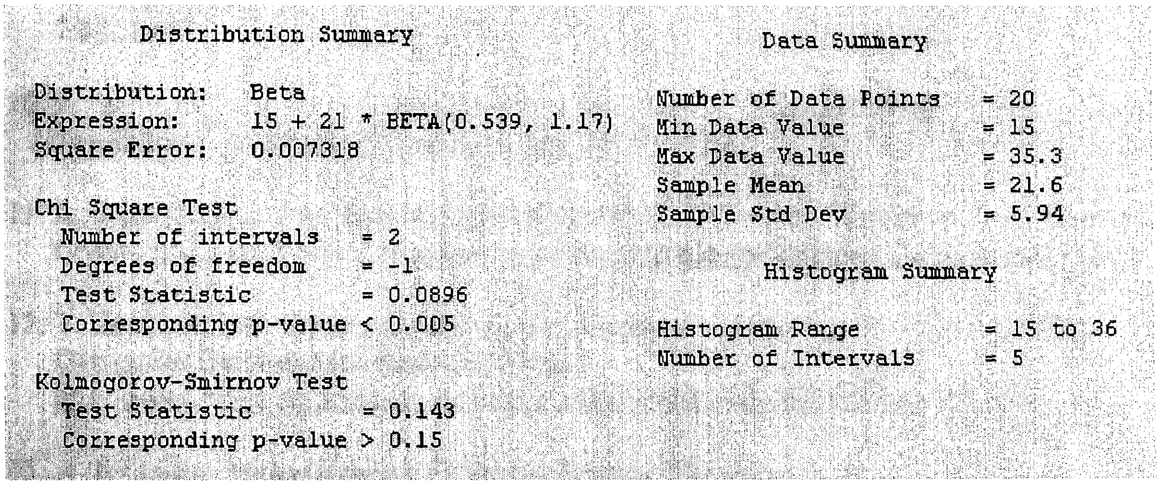
Figure 5 Distribution Fit for Monitor

APPENDIX (Cont.)

Table 5 Disassembly Times in Minutes for Monitor

Televisions							
Disassembly Times in Minutes	27.5	31.45	35.25	16.53	19.9	20.3	24.35
	18.2	18.3	30.25	15.02	20.2	15.5	16.24
	18.5	16.25	18.09	25.3	27.05	18.55	





```
        Distribution Summary                        Data Summary

Distribution:    Beta                      Number of Data Points    = 20
Expression:      15 + 21 * BETA(0.539, 1.17)  Min Data Value        = 15
Square Error:    0.007318                  Max Data Value           = 35.3
                                           Sample Mean              = 21.6
Chi Square Test                            Sample Std Dev           = 5.94
  Number of intervals    = 2
  Degrees of freedom     = -1                    Histogram Summary
  Test Statistic         = 0.0896
  Corresponding p-value < 0.005            Histogram Range          = 15 to 36
                                           Number of Intervals      = 5
Kolmogorov-Smirnov Test
  Test Statistic         = 0.143
  Corresponding p-value > 0.15
```

Figure 6 Distribution Fit for Printer

REFERENCES

1. D. T. Allen and K. S. Rosselot, *Pollution Prevention for Chemical Processes,* John Wiley & Sons, New York, 1997.

2. R. M. Koerner and D. E. Daniel, *Final Covers for Solid Waste Landfills and Abandoned Dumps,* American Society of Civil Engineers, Virginia, 1997.

3. M. Pidd, *Computer Simulation in Management Science,* John Wiley & Sons, New York, 1984.

4. W. D. Kelton, R. P. Sadowski and D. A. Sadowski, *Simulation with Arena,* WCB McGraw-Hill, 2nd edition, New York, 1997.

5. J. Banks and J. S. Carson, II, *Discrete-event System Simulation*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

6. S. V. Hoover and R. F. Perry, *Simulation: A Problem-Solving Approach,* Addison-Wesley Publishing Company, Inc., 1989.

7. J. Ji, "Multi-lifecycle Assessment Design Tools and Software Development", Masters Thesis for Computer Engineering, New Jersey Institute of Technology, Newark, NJ, January 1999.

8. R. G. Sargent, *Validation and Verification of Simulation Models*, Winter Simulation Conference Proceedings, pp. 39-55, 1999.

9. J. Banks, *Introduction to Simulation*, Winter Simulation Conference Proceedings, pp. 7-13, 1999.

10. A. M. Law and M. G. McComas, *Simulation of Manufacturing System*, Winter Simulation Conference Proceedings, pp. 57-59, 1999.

11. A. Jean-Jacques, *The Remanufacturing Process,* Design for Environmental Research Group, http://sun1.mpce.stu.mmu.ac.uk/pages/projects/dfe/pubs/dfe24/report24.htm

12. T. Dowie and M. Simon, *Guidelines for designing for disassembly and recycling*, Design for Environment Research Group, http://sun1.mpce.stu.mmu.ac.uk/pages/projects/dfe/pubs/dfe18/report18.htm

13. A. Bylinsky, *Manufacturing for reuse, Fortune Magazine*, February 6, 1995.

14. R. J. Caudill, *Proposal to Establish the Multi-lifecycle Engineering Research Center,* A Final Proposal to National Science Foundation: Engineering Research Center Program, 1997.

15. R. D. Hurrion, *Simulation: Application in Manufacturing*, IFS (Publications) Ltd. New York, 1986.

16. T.C. Goodhead and T.M. Mohoney, *Experience in The Use of Computer Simulation for FMS Planning*, John Wiley & Sons, New York, 1998.

17. N.R. Greenwood, P. Rao and M. Wisnom, *Computer Simulation for FMS*, John Wiley & Sons, 1st edition, New York, 1998.

18. L. Ketan, *System Simulation and Modeling of Electronics Demanufacturing Facility*, Masters Thesis for Multi-Lifecycle Engineering Research Center, Mechanical Eng. Dept, New Jersey Institute of Technology, Newark, NJ, December 1998.

19. Y. Tang, M.C. Zhou, and R. J. Caudill, *An Integrated Approach to Disassembly Planning and Demanufacturing Operation*

20. A. Gungor and S. M. Gupta, *Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey*, Laboratory for Responsible Manufacturing, Northwestern University, Boston, MA.

21. M.C. Zhou, R. J. Caudill and D. Sebastian, *Multi-lifecycle Product Recovery for Electronic Products*, Journal of Electronics Manufacturing, Vol. 9, No 1, pp. 1-15, March 1999.

22. E. Zussman and M.C. Zhou, *Design and Implementation of an Adaptive Process Planner for Disassembly Processes*, IEEE Transactions on Robotics and Automation, Vol. 16, No. 2, April 2000.

23. F. Jovan et al., *A Key Issue in Product Life-cycle: Disassembly*, Annual of CIRP, Vol. 42/2, 1993.

24. P. Kopacek and G. Kronreif, *Semi-automated Robotized Disassembling of Personal Computers.* IEEE Conference on Emerging Technologies and Factory Automation, Nov. 18-21, pp. 567-572, Kauai, Hawaii, 1996.

25. K. Umeda and T. Arai, *Three-Dimensional Vision System for Mechanical Assembly/Disassembly.* Advanced Robotics, 11(2), pp. 147-167, 1997.

26. E. Zussman and M.C. Zhou, *A methodology for Modeling and Adaptive Planning of Disassembly Processes*, IEEE Trans. Robot. Automat., Vol. 15, pp. 190-194, Feb. 1999.

27. Homen de M. L.S. and Sanderson A. C., *AND/OR Graph Representation of Assembly Plans*, IEEE Trans. On Rob. & Aut., Vol.6. No.2, 1990, pp.188-199.