12-2004

Implementation of Critical Path Method and Critical Resource Diagramming Using Arena Simulation Software

Jamie Ehresman Gunter
University of Tennessee, Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Jamie Ehresman Gunter entitled "Implementation of Critical Path Method and Critical Resource Diagramming Using Arena Simulation Software." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Adedeji B. Badiru, Major Professor

We have read this thesis and recommend its acceptance:

Dukwon Kim, Dong Joon Kong

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Jamie Ehresman Gunter entitled "Implementation of Critical Path Method and Critical Resource Diagramming Using Arena Simulation Software." I have examined the final paper copy of this thesis for form and content and recommend that it may be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Badiru

Adedeji B. Badiru, Major Professor

We have read this thesis
and recommend its acceptance:

Dukwon Kim

Dongjoon Kong

Accepted for the Council:

Vice Chancellor and
Dean of Graduate Studies

IMPLEMENTATION OF CRITICAL PATH METHOD AND CRITICAL RESOURCE DIAGRAMMING USING ARENA SIMULATION SOFTWARE

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Jamie Ehresman Gunter

December 2004

DEDICATION

This thesis is dedicated to my parents, Howard Ehresman and Christine Ehresman, great role models and friends, and my brothers, Adam Ehresman and Howard Ehresman, and my husband, Allen Gunter, for always believing in me, inspiring me, loving me, and encouraging me to reach higher in order to achieve my goals.

Acknowledgments

I wish to thank all those who helped me to complete my Master of Science degree in Industrial Engineering. I would especially like to thank Dr. Adedeji Badiru for his guidance and support through my education experience. His valued advice, guidance, and encouragement were influential in completing this thesis.

I would also like to thank Dr. Dukwon Kim and Dr. DongJoon Kong for serving on my committee and guiding me with great advice.

Appreciation is in order for Ammar Aamer for reviewing Arena with me and guiding me through the simulation portion.

I would like to thank my family and friends whose suggestions, encouragement, love, and support kept me going through school and the hard times in life.

Lastly I wish to thank my husband for putting up with me, supporting me, and still loving me through this stressful time. Without his love and support this thesis would never have been possible.

Abstract

A relatively new resource management tool, which considers both time and resource requirements, is Critical Resource Diagramming (CRD). CRD is a simple extension to the CPM technique developed for resource management purposes. It is a graphical tool used mainly for resource scheduling.

The purpose of this study is to use the Arena simulation software and CRD to solve problems. This will be accomplished by setting up an example problem in the simulation program, Arena, and the CRD approach will be used as a deterministic and probabilistic problem.

The Arena output results for the deterministic parts of the example were compared to the output that was done by hand and the values obtained were exactly the same. Thus, Arena can be used as an effective and accurate management tool for resource scheduling.

The probabilistic results of the Arena output show that if the user does not know the exact time of the process, that a distribution can be used to give results. All the user would have to know is the approximate process time, what distribution the process times will follow, and the standard deviation of the distribution.

Arena is both easy and a user-friendly simulation software, thus providing a very simple and important tool in the area of project management. In summary, the research reported here has made a significant contribution in enhancing the CPM method and applying the CRD method to a new application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

LIST OF ABBREVIATIONS

a	optimistic PERT time estimate
ACTIM	Activity Time
ACTRES	Activity Resource
AOA	activity-on-arrow
AON	activity-on-node
b	pessimistic PERT time estimate
CAF	Composite Allocation Factor
CPM	Critical Path Method
CRD	Critical Resource Diagramming
DARN	Dual Allocation of Resource Networks
EC(*i*)	Earliest completion time for activity *i*
ES(*i*)	Earliest start time for activity *i*
i	Activity identification
LC(*i*)	Latest completion time of activity *i*
LS(*i*)	Latest state time of activity *i*
m	mostly likely PERT time estimate
P(*i*)	Set of immediate predecessors of activity *i*
PERT	Project Evaluation Review Technique
R-R	Resource Relationship
RAF	Resource Allocation Factor
ROT	Resource Over Time

S(*j*)	Set of immediate successors of activity *j*
SAF	Stochastic Allocation Factor
t	Duration of activity *i*
T_p	Latest allowable occurrence time for a project node

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In the area of project management, the topic of prominent research has been the problem of resource constrained scheduling. Scheduling rules have been classified as two main types, optimization techniques and scheduling heuristics. As reported by Davis and Patterson (1975), optimization techniques, though accurate, seem to be limited only to unrealistically small projects. On the other hand, scheduling heuristics have been reported inconsistent but easy to use and applicable to a variety of projects.

Many methods, both old and new have been applied using many different tools and applications. These attempts tried to optimize the results of resource scheduling, while finding a consistent way to schedule resources, but nothing seems to consistently give a "great" result. One such method is Critical Path Method (CPM). This method works ok, but does not show how much of the resources are being used. Therefore, this research is going to focus on Critical Resource Diagramming (CRD) and compare it to the CPM method.

1.2 Proposed Methodology

A hypothetical sample problem is going to be created and solved using both CPM and CRD. A simulation application is then going to be applied to this sample problem. Simulation is a powerful tool if understood and used properly (Ingalls 2002). According to Robert E. Shannon (1975), simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies for the operation of the system. Rockwell Software offers advanced simulation software with Arena 5.0. According to Collins and Watson (1993), Arena offers a high level of modeling flexibility across a wide range of problem domains, yet is very simple to learn and use. The Arena simulation software will be used to analyze and compare CPM and CRD, their results, and then to hopefully improve their results and conclusions and produce another and hopefully better way to schedule activities and resources.

1.3 Objectives of Research

The main objective of this research is to use a relatively new resource management method called Critical Resource Diagramming (CRD) and the very popular Critical Path Method (CPM), and to apply both these methods to an application that has never been used before for these techniques. The second objective is to try to improve the effectiveness of project and resource scheduling when applying this new application to CPM and CRD. The next purpose is to find a new and better way to schedule resources effectively, without affecting the time duration of the project. Finally, the last objective of this research is to show the differences in the CPM and CRD methods and the results obtained when using these methods to solve the same problem.

1.4 Thesis Organization

The remaining portions of Chapter 1 will focus on the origin of the methods that will be used for this research. Chapter 2 focuses on previous methods and applications that have been used to try to solve resource-scheduling problems. Chapter 3 discusses the fundamentals of the methods that are used and how they will be applied to this research. Chapter 4 concentrates on the tool that will be applied to the methods of consideration. Included in Chapter 4 are the hypothetical example problem, its setup, and the results. Chapter 5 is the concluding chapter and also contains ideas for further studies using this research.

1.5 Origin of Research

Project scheduling is the time-phased sequencing of activities subject to precedence relationships, time constraints, and resource limitations to accomplish specific objectives as defined by Badiru and Pulat (1995). Project scheduling differs from flow shop, job shop, and other production sequencing problems due to the non-repetitive nature of the activities involved in a project. The manufacture and assembly of a large ski boat, the installation of a heating and air system, and the construction of a new house or road are typical examples of a project.

The two main techniques of project management useful in the basic managerial functions of planning, scheduling, and control are Critical Path Method (CPM) and Project Evaluation and Review Technique (PERT). As reported by Wiest and Levy (1969), the planning phase of any venture involves the listing of tasks or jobs that must be performed to bring about the venture's completion. Also in this phase, the resource requirements are determined and the cost and job duration estimates are made. The laying out of the actual tasks of the project in the time order in which they have to be performed is scheduling (Wiest and Levy 1969). The expected completion time of each task and the personnel and material requirements needed at each stage of production are calculated. Control begins with the review of the differences between the schedule and the actual performance once the project has begun (Wiest and Levy 1969). CPM and PERT are useful tools that utilize all three of the above functions.

1.5.1 Critical Path Method (CPM)

Initiated in 1957 by the DuPont Company along with Remington Rand, Critical Path Method (CPM) has become widely used, especially in the construction and process industries (Kerzner, 2003). CPM is basically concerned with obtaining the trade-off between cost and completion dates for large projects. CPM emphasizes the relationship between applying more resources or increased cost as mentioned by Wiest and Levy (1969). The amounts of time needed to complete various aspects of the project are assumed to be known with certainty when applying the CPM method. Also the relationship between the amount of resources employed and the time needed to complete the project is assumed known. Therefore, as stated by Whitehouse and Brown (1979), CPM is a deterministic model that basically deals with the time-cost trade-off.

1.5.2 Project Evaluation and Review Technique (PERT)

In 1958 and 1959, to meet the needs of the "age of massive engineering," Program Evaluation and Review Technique (PERT) was originally developed and used where the techniques of Taylor and Gantt were inapplicable. In 1958, with the aid of the management-consulting firm of Booz, Allen, and Hamilton, PERT was introduced by the Special Projects Office of the U.S. Navy, on its Polaris Weapon System. Since that time, PERT has spread rapidly throughout almost all industries (Kerzner, 2003). PERT is a mixture of new ideas and proven techniques. As stated by Kerzner (2003), it can be considered as a road map for a particular project in which all the major elements (events) have been completely identified, together with their corresponding interrelations. PERT takes into account the uncertainties in the activity duration time estimates. It assumes that the activities and their network relationships have been well defined, but it allows for uncertainties in activity times. Not only is an estimate made of the most probable time required to complete the activity (denoted by "m") for each activity in the network, but some measure of uncertainty is also noted for this estimate. The other two time estimates for the PERT procedure are the pessimistic estimate, denoted by "b," and the optimistic estimate, denoted by "a." These three time estimates are given by the person who would be most qualified to know – an engineer, supervisor, or worker.

1.5.3 Critical Resource Diagramming (CRD)

Developed by Badiru (1992), critical resource diagramming (CRD) is an important resource management tool. It is a simple extension to the CPM technique developed for resource management purposes. CRD is a graphical tool, which brings the advantages of CPM to resource scheduling. CRD can be used to develop strategies for assigning activities to resources and vice versa. The results of CRD are used in simplified resource tracking and control, better information to avoid job conflicts, and better job distribution.

1.6 Resource Limitations in Project Scheduling

There are several reasons to consider resource allocation in project scheduling. One reason is to avoid inherent inconsistencies. An example would be using the same resource on two tasks at the same time. Another reason for resource allocation is that the forecasted use of some key resources may indicate that there will be a surplus of resources sometime in the future.

The basic assumption of both the CPM and PERT technique is that the resources required to perform the activities are available in unlimited supply, or that at least there is enough resources available for each activity that is scheduled. Sometimes, estimates are influenced by a consideration of available resources for individual activity times. But such estimates are usually made independently of the other activity estimates, and the same resources are not considered for possible competing claims. Activities that occur on parallel paths through the network may use the same resources and even though precedence constraints would not prevent their being scheduled, a limited supply of resources might force them to be scheduled sequentially.

There are three distinct categories when considering the types of resource allocation, as suggested by Davis (1973). These categories are:

1. Time-cost trade-off problems: involves the determination of the least-cost schedule for any project duration. It arises in the context of long range resource planning, where management seeks to determine the combination of resource levels and project due dates that will minimize resource costs, overhead costs, and losses which result when project deadlines are not met (Moder and Phillips 1970). The procedure assumes infinite availability of resources.
2. Resource leveling problems: arises when it is possible to procure sufficient resources to carry out a project which must be completed by a specified due date. The resource leveling procedure attempts to reduce any fluctuations in the level of resource usage while maintaining project duration. This project duration is the

duration as calculated by critical path procedures. Activities are then rescheduled within their available slack to give the most acceptable profile of resource usage over time.

3. Limited resource allocation problems: arises when the amount of resources available during a project is not sufficient to satisfy the demands of concurrent activities. Sequencing decisions are used to determine which of the competing activities receive the priority for resource allocation in an attempt to find the shortest project schedule consistent with the fixed resource limits. The procedures available are of three types (Jayaraman 1995): analytical, analogue, or heuristic.

CHAPTER 2

REVIEW OF LITERATURE

Resource allocation in project scheduling is receiving more attention today than it has in the past. Several reasons for this could be due to the fact that the significance of the problem is growing rapidly. Modern technology has developed many large resources that must be accounted for. The fact that the number of personnel resources is increasing from the increase in the number of different trades which can further be broken down by geographic barriers, skills, departments, etc. is also an important reason. With these reasons, it is not uncommon to deal with problems where large number of resources must be considered.

Also, the interest in finding an optimal solution to schedule projects is continuous. If we were to consider all the scheduling possibilities, a combinatorial problem of difficult magnitude would be created. Since there is not a mathematical basis for a realistic scheduling procedure, the development of an optimal solution does not appear likely in the near future.

In the past, there have been a few attempts to approach resource scheduling with the use of simulation and its softwares. The important task is to develop a simulation model whose underlying assumptions are more widely accepted. The focus of this research is to make simulation, particularly Arena, more widely accepted for resource scheduling problems.

2.1 Optimization Techniques

There are many techniques that use the optimal approach to solve resource-scheduling problems even though they are computationally exhaustive and impractical to use, because the problems end up becoming too complex. Most of the early research in this area concentrated on the formulation and solution of the problem as an integer-programming problem. Early attempts to solve the exact version integer-programming problem were unsuccessful (Davis 1973). This caused many researchers to therefore develop numerous enumerative approaches for the solving of certain variants of this problem optimally.

The basic network schedule problem can be viewed as an integer-programming problem (Charnes and Cooper 1962). Their approach views a project network as a flow network in which a hypothetical unit of flow leaves the source node and enters the sink node. Fulkerson (1961) formulated a general network flow theory to solve the project network time-cost trade-off problem. The procedure involves computing the maximum possible flow (steady-state flow rate) from source to sink. Both these approaches formed linear programming problems. Charnes and Cooper's (1962) method was used to solve the duality theorem of linear programming. Fulkerson's (1961) tactic maximized the total network flow, which is equal to the total flow entering the sink node.

A Branch and Bound procedure was developed by Stinson et al. (1978) to solve the multiple resource constrained project-scheduling problem. The authors use a decision vector, which consists of a series of tie-breaking rules for selecting the next partial schedule to branch to, or the next candidate problem to consider. The procedure establishes a new search origin for candidate problem selection. The authors declare that their procedure was successful in obtaining an optimal solution for 97% of the problems attempted.

Talbot and Patterson (1978) present an integer-programming algorithm for allocating resources to competing activities of a project such that the completion time of the project is minimal along all possible completion times. The systematic enumeration of all possible job finish times for each task of the project was needed to achieve this end. A biasing method called Network Cut was developed to remove from consideration the evaluation of job finish times, which cannot lead to a reduced project completion time. The authors state that the algorithm is very reliable for projects consisting of 30-50 activities and 3 different resource types. The authors concluded that their procedure is more efficient than other implicit enumeration procedures, and is comparable with the best available Branch and Bound technique.

Johnson (1992) benchmarks the project duration computed by Talbot and Patterson (1978) because of their reported high efficiency rate and reliability in finding the optimal solution. Johnson (1992) used their computed project duration to test the accuracy of the computer software, STARC, and his results are in agreement with those of Talbot and Patterson (1978).

Numerous other enumerative approaches for solving certain variants of the optimization problem have also been developed (Balas 1970, Bell and Park 1990, Christofides et al. 1987, Davis 1969, Davis and Heidorn 1971, Fisher 1973, Gorenstein 1972, Patterson and Huber 1974, Patterson and Roth 1976, Schrage 1970, Talbot 1976 and 1982).

2.1 Heuristic Techniques

Because of the lack of success with optimization procedures, major efforts in attacking the problem have been expended in developing heuristic techniques. These heuristic techniques seem to produce "good" feasible solutions. The "heuristics" or rules used in obtaining such solutions are schemes for assigning activity priorities in making activity-sequencing decisions required for resolution of resource conflicts (Davis, 1975). There exist two types of heuristic programs for resource scheduling:

1. Resource leveling programs: The scheduling objective is to attempt to eliminate the manpower peaks and valleys by smoothing out the period-to-period resource requirements. The ideal situation is to do this without changing the end date.
2. Resource allocation programs: The scheduling objective is to attempt to find the shortest possible critical path based upon the available or fixed resources.

2.2.1 Resource Leveling Programs

Early work on resource leveling was carried on concurrently by Burgess and Killebrew (1962), and Levy, Thompson, and Wiest (1962). Both teams developed a computer system for leveling project resources.

Burgess and Killebrew (1962) developed a method, which uses the sum of squares as a measure of scheduling effectiveness. These authors found the number of activities in progress in a time period and squared this number. They then added the squares for all periods in the project. The smaller the sum and the lower the resource profile, offers the better scheduling effectiveness. The authors then use this information to schedule the activities in order to achieve this end.

Levy et al. (1962) developed a procedure called MS^2 (multi-ship, multi-shop) for workload smoothing, designed originally to smooth work force requirements in naval shipyards. This procedure starts with an early start schedule and profile for the project and then identifies activities in progress during peak period(s). An activity that has adequate slack to shift beyond the peak is randomly selected and scheduled beyond the peak. A new resource peak is identified and the process is repeated until the peak can no longer be reduced.

For resource leveling problems with special objective functions and only special minimum time lags, several exact and heuristic solution procedures have been proposed (Neumann 2000). Exact algorithms based upon (implicit) enumeration, integer programming, or dynamic programming techniques have been devised by Ahuja (1976), Easa (1989), Bandelloni et al. (1994), Demeulemeester (1995), and Younis and Saad (1996). More recently, Savin et al. (1996, 1997) have proposed a neural network approach for solving resource-leveling problems.

2.2.2 Resource Allocation Programs

About ninety percent of the projects dealt in practice are resource constrained projects (Jayaraman 1995). Therefore, most of heuristics concentrate on solving the resource constrained problem.

One of the earliest heuristics developed in resource-constrained project scheduling is the Brook's Algorithm (Brooks and White 1965), presented by Bedworth

(1982). This method uses a sequencing criterion called ACTIM (Activity Time). The ACTIM signifies the maximum time that an activity controls in the project on any one path. ACTIM is represented by the following equation and is scaled from 0 to 100:

$$\text{ACTIM} = (\text{critical path time}) - (\text{activity latest start time})$$

A few years later, Bedworth (1973), proposed a scheduling heuristic called ACTRES (Activity Resource), and a priority rule named TIMRES (Time-Resources). ACTRES is a combination of an activity time and resource requirements and is represented by the following equation:

$$\text{ACTRES} = (\text{activity time}) * (\text{resource requirement})$$

TIMRES is composed of equally weighted portions of ACTIM and ACTRES and is expressed as:

$$\text{TIMRES} = 0.5(\text{ACTIM}) + 0.5(\text{ACTRES})$$

GENRES is anther extension to the Brooks algorithm proposed by Whitehouse and Brown (1979). GENRES is a modification of TIMRES with weighted combination of ACTIM and ACTRES. Weights (w) between 0 and 1 are used in the following expression:

$$\text{GENRES} = w(\text{ACTIM}) + (1-w)(\text{ACTRES})$$

Badiru (1988c) compares the above mentioned heuristics using standard test problems. He also presents a performance measure to compare the various project scheduling heuristics.

Many other resource allocation techniques have been proposed over the years. One such proposal is the ROT (Resource Over Time) equation, which is represented by the following expression (Elsayed 1982):

$$ROT = \frac{\mathit{RESOURCE\ REQUIREMENT}}{\mathit{ACTIVITY\ TIME}}$$

Another method for scheduling large resource constrained projects called SPAR-1 allocates available resources, period by period, to project activities listed accordingly to their earliest start times (Wiest 1967). Another such technique called the Blob Chart, similar to the well known Gantt Chart, was developed by Beimborn and Gavey (1972), but has been declared limited to extremely small projects. One last example of other methods developed in previous years is a graphical tool called DARN (Dual Allocation of Resources in Networks), which uses the topological dual of the activities-on-arcs networks, and was found capable of providing very reliable heuristic solutions to very small projects (Biegel and Halim 1975).

2.3 Simulation Project Networks

Simulation has been used to study such wide ranging topics as urban systems, economic systems, business systems, biological systems, production systems, social systems, transportation systems, health care delivery systems, and many more. As stated by Pritsker (1986), simulation is the most widely used management science and operations research technique employed by industry and government. Nonetheless, as stated by Chan (1997), the use of simulation programs to solve project-scheduling problems is rare. However, simulation is becoming an important analytical tool in project planning, while also becoming one of the most important approaches in the analysis of project networks.

Badiru (1984) developed a resource scheduling heuristic called STARC, which is used to simulate project networks and perform what-if analysis of projects involving probabilistic activity times and resource constraints. STARC uses Monte Carlo simulation to generate the activity times from a known beta distribution. The simulated schedules outputted by the program are planned to serve as decision aids for the project managers. Outputs of simulation can be used to alert management to impending and potential bottlenecks.

Jayaraman (1995) extended on Badiru's STARC program and incorporated the use of a goal programming to determine the weight between Resource Allocation Factor (RAF) and Stochastic Activity Factor (SAF). Also incorporated into this program was the use of critical resource diagramming (CRD), which was used to calculate the risk factor. A new scaling approach was implemented to calculate the value of Composite Allocation Factor (CAF) from RAF and SAF. The output of Jayaraman's research was compared with other various commercial software and the values obtained were very encouraging.

More recently, some simulation softwares have been used to test the methods of research. For example, in 2002, Simmons used the ProcessModel simulation software along with CPM and PERT. As stated by Simmons (2002), ProcessModel simulation allows the activity times of a project to be represented by a variety of distributions and further the resulting project time may also be represented by a variety of distributions.

CHAPTER 3

FUNDAMENTALS OF NETWORK ANALYSIS

A network diagram is a graphical tool used to represent the contents and the objectives of the project. The network is a graphical flow diagram composed of events and activities. The network shows the interrelationships, interdependencies, and the sequence of all the activities and events that must be completed in order to conclude the project.

The first step in project scheduling is to sketch an arrow diagram, which is used to represent and show inter-dependencies and the precedence relationships among the activities of the project. As stated by Jayaraman (1995), an important function of a network diagram is to provide a comprehensive picture of the precedence relationships among activities. By comparing the environments created by the completion of an individual activity in the project with the environment necessary to start the succeeding activities, precedence can be determined. When determining precedence relationships among activities, resource availability should not be considered. The two most common network models are the *activity-on-arrow (AOA)* and the *activity-on-node (AON)* conventions. The AOA approach uses arrows to represent the activities and uses the nodes to represent the starting and ending points of the activities. In the AON method, nodes are used to represent the activities and the arrows are used to represent precedence relationships. The AON approach was used for this research.

3.1 Activity-On-Node (AON) Diagram

When constructing a network presentation of a project, certain symbols and basic definitions are used. The following dialogue presents the major network terms and their symbols.

1. Node – a circular representation of an activity. O

2. Arrow – a line having an arrowhead at one end used to connect two nodes. Also used to represent the activity at the end of the arrow as the next step after the preceding node is completed. →

3. Activity – the work required to proceed from one event or point in time to another or any portion of the project, which consumes time and resources while having a definable beginning and ending. In the AON approach, the activity is represented by a node and the task the activity represents may be indicated by a symbol inside the node. Activities can be classified into the following five categories:

   (i) Predecessor activity: the node that immediately precedes the activity being considered. It must be completed immediately prior to the start of another activity. In Figure 3.1, A is a predecessor of B and both B and C are predecessors of D.

   (ii) Successor activity: the activity that immediately follows the one being considered. In Figure 3.1, activities B and C are successors to activity A.

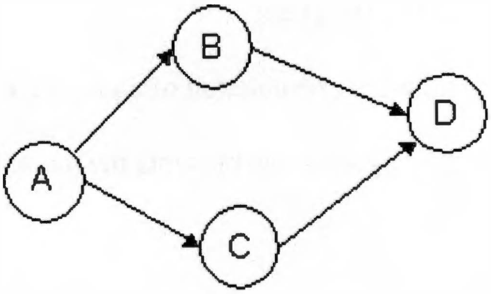


Figure 3.1 A Project Network (AON)

(iii) Descendent activity: any activity that is restricted by the one in consideration. Activities B, C, and D are all descendents of activity A in Figure 3.1.

(iv) Antecedent activity: any activity that must precede the one under consideration. In Figure 3.1, activities A and C are antecedents of activity D. Activity A is also an antecedent of B, but activity A has no antecedent.

(v) Dummy activity: an activity that does not consume any resources or time but simply portrays a technological dependence. It is not required but may be included for network clarification, convenience, or to represent an achievement in the progress of the project. It is represented by a dashed circle and treated as an activity with zero time duration.

4. Event – the starting or ending point for a group of activities. Events can be further classified into the following two categories:

   (i) Merge Event: occurs when two or more activities are predecessors to a single activity as shown in Figure 3.2. All the preceding activities (A, B, and C) must take place before the merge activity (D) can transpire.

   (ii) Burst Event: exists when two or more activities have a common predecessor as shown in Figure 3.3. The activities following the burst event (A) cannot be started until the burst event has been completed.

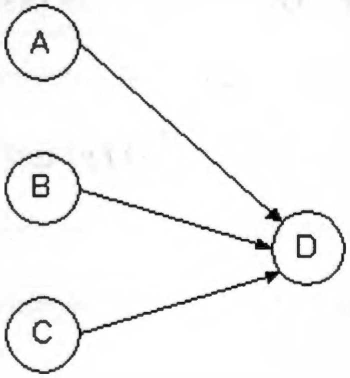


Figure 3.2 Merge Event

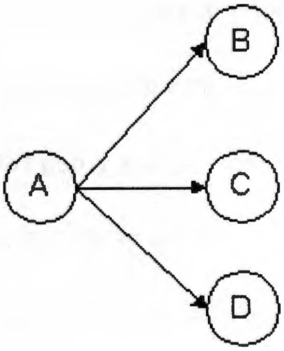


Figure 3.3 Burst Event

3.2 CPM Procedure

Once the network of a project is constructed, the basic result in CPM is the construction of the time schedule for the project and its ultimate objective is to determine the project's critical path. The critical path is the longest path in the project network. In other words, it is the sequence of project activities that determines the minimum completion time for the project. The time analysis also becomes an important part of the process for the planning of various activities. An activity time is the forecasted time an activity is expected to take from its starting point to its completion point. The other objectives of the CPM procedure are to find:

1. Total completion time of the project.
2. Earliest time when each activity can start.
3. Latest time when each activity can be started without delaying a project.
4. Categorization of the activities of the project as critical and non-critical.

3.2.1 Basic Computation Classification

The basic computations in CPM first involve a forward pass and then a backward pass through the network. The following notations are used in this research to represent the basic scheduling computations.

i: Activity identification

ES(*i*): Earliest start time for activity *i*

LS(*i*): Latest start time for activity *i*

EC(*i*): Earliest completion time for activity *i*

LC(*i*): Latest completion time for activity *i*

t: Duration of activity *i*

The graphical representation of each node representing the various start and completion times is shown in Figure 3.4

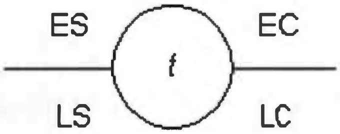


Figure 3.4 CPM Computation Representation

3.2.2 Forward Pass Computation

The forward pass is used to determine the earliest start and earliest completion times for each activity in the project. In the forward pass, the computations start at the start node, node 1, and advance recursively to end node ***n*** using the following steps:

1. Set the starting time of the project to be zero. In other words, set the earliest start time for node 1 as zero.

$$\mathrm{ES}(i) = 0$$

2. Calculate the earliest start time (ES) for any node (activity j) as the maximum of the earliest completion times (EC) of the predecessors of the node.

$$\mathrm{ES}(i) = \max_{j \in P(i)} \{\mathrm{EC}(j)\}$$

where P(i) is the set of immediate predecessors of activity i.

3. Calculate the earliest completion time (EC) as the sum of the activities earliest start time and the activity duration time, t_i.

$$\mathrm{EC}(i) = \mathrm{ES}(i) + t_i$$

The earliest completion time of the project is equal to the earliest completion time of the very last node, node ***n***, in the project network.

$$\mathrm{EC}(\mathit{Project}) = \mathrm{EC}(n)$$

3.2.3 Backward Pass Computation

The backward pass is used to calculate the latest start and latest completion times of an activity. These computations are a “mirror image” of the forward pass computations. The backward pass computation starts at the last node, node *n*, and regress recursively back to the start node, node 1 using the following steps:

1. Set the latest completion time of the project. Using the latest allowable occurrence time, T_p, for the project node, can do this.

   $$\mathrm{LC}(\textit{Project}) = T_p$$

   If no scheduled completion date for the project is specified, then it is set equal to the earliest completion time of the project.

   $$\mathrm{LC}(\textit{Project}) = \mathrm{EC}(\textit{Project})$$

2. Calculate the latest completion time for activity *j* as the minimum of the latest start time of the activity's immediate successors.

   $$\mathrm{LC}(j) = \min \{\mathrm{EC}(j)\}$$

   $$i \in S(\mathrm{j})$$

   where S(*j*) is the set of immediate successors of activity *j*.

3. Calculate the latest start time for activity *j* as the subtraction of the latest completion time and the activity duration time.

   $$\mathrm{LS}(j) = \mathrm{LC}(j) - t_i$$

3.2.4 Determination of Critical Path

The critical path is the path with the least total slack or the path with no slack at all. It is the longest path in the network diagram and defines the minimum time required to complete the project. An activity will be on critical path if the earliest and latest occurrence times of the end events are equal and if the duration occupies their time span exactly. The critical activities of a network must constitute an uninterrupted path that spans the entire network from start to finish. An activity is defined as critical when there is no "flexibility" in determining its start and finish times. In other words, each critical activity must be started and completed on time, to complete the project without delay. A non-critical activity is one that allows some scheduling slack, meaning that the start time of an activity may be advanced or delayed within limits without affecting the completion date of the entire project.

The critical path can be determined in CPM analysis by the forward pass only. As reported by Badiru and Pulat (1995), the following steps are used to determine the critical path:

1. Complete the forward pass.
2. Identify the last node in the network as a critical activity.
3. If activity *i* is an immediate predecessor of the critical activity *j*, and if EC(*i*) = ES(*j*), then activity *i* is a critical activity. When all the immediate predecessors of activity *j* are considered mark activity *j*.
4. Continue backtracking from each unmarked critical activity until the project-starting node is reached.

3.3 PERT Procedure

As discussed so far, it is implicitly assumed that in the CPM procedure the time values are deterministic. This assumption is valid in regular activities such as maintenance of machines, automated systems, construction of a house, etc. However, in research projects or design of new machinery, various activity times are based on judgment.

PERT is the procedure that takes into account uncertainties in activity duration. The PERT approach takes into account three time values associated with each activity: the optimistic time, the pessimistic time, and the most likely time. These three time estimates provide a measure of uncertainty associated with each activity.

1. The optimistic time signifies the shortest time in which an activity can be completed. It is denoted by a, and assumes that everything goes well. This estimate is unrealistic in that it is expected to occur in one case out of fifty (Jayaraman 1995).
2. The pessimistic time represents the longest completion time of an activity. It is denoted by b, and assumes that everything that could logically go wrong does go wrong. This is an unrealistic estimate in that it is expected to occur in one case out a hundred (Jayaraman 1995).
3. The most likely time is the estimate of the normal time it would take the activity to be completed. It is denoted by m ($a < m < b$), and assumes normal delays.

3.4 CRD Procedure

CRD is a simple extension to the CPM technique developed for resource management purposes. CRD is a graphical tool whose use results in simplified resource tracking and control, better job distribution, and better information to avoid resource conflicts.

3.4.1 CRD Network Development

In CRD, each node is used to represent each resource unit. Each node, j, in the project network refers to the task responsibility of resource type j. The resources are interrelated using resource-relationship (R-R) arrows. The CRD method uses the same precedence requirements as in the CPM procedure. The CRD network for a project that consists of three different resource types is shown in Figure 3.5. There are four nodes in Figure 3.5, as there are two units of resource type 3.

3.4.2 CRD Classifications

In a CRD network the following terms are used:

1. Bottleneck Resource: a node at which two or more arrows merge (Badiru 1992).
2. Dependent Resource: a node whose task depends on the task of immediately preceding nodes (Badiru 1992).
3. Critically Dependent Resource: a node on the critical resource path at which several arrows merge (Badiru 1992). An example of this could be node Res 2 from Figure 3.5.

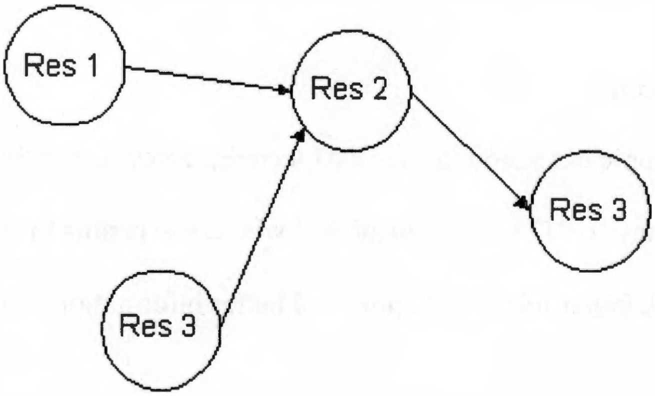


Figure 3.5 CRD Network

3.4.3 CRD Computations

In CRD, the same forward and backward passes of CPM are applicable. On the other hand, the interpretation of the critical path may be different due to the fact that a single resource may appear in multiple nodes.

CHAPTER 4

SIMULATION AND ARENA

Simulation is becoming an important and popular tool for the use of project scheduling. Simulation can predict with very precise accuracy if used correctly, which is one reason it is becoming such a popular analysis tool. Arena is a widely used simulation software that is very easy to learn, therefore making it very user-friendly.

4.1 Simulation

According to Kelton et al. (2004), simulation refers to a broad collection of methods and applications to mimic the behavior of real systems, usually on a computer with appropriate software. Computer simulation refers to methods for studying a wide variety of models of real world systems by numerical evaluation using software designed to imitate the system's operations or characteristics, often over time (Kelton et al. 2004). Simulation models can be employed at four levels (Pritsker, 1986):

- As explanatory devices to define a system or problem;
- As analysis vehicles to determine critical elements, components, and issues;
- As design assessors to synthesize and evaluate proposed solutions;
- As predictors to forecast and aid in planning future developments.

Simulation is the process of designing and creating a computerized model of a real or proposed system for the purpose of conducting numerical experiments to give us a better understanding of the behavior of the system for a given set of conditions (Kelton et al. 2004).

There are many situations in which the real-world system becomes too complex to be expressed in simple equations that can be solved in a reasonable amount of time (Chan 1997). Simulation experiments are very useful in that they allow inferences to be drawn about systems (Pritsker, 1986):

- Without building them, if they are only proposed systems;
- Without disturbing them, if they are operating systems that are costly or unsafe to experiment with;
- Without destroying them, if the object of an experiment is to determine their limits of stress.

Simulation does not necessarily always give the optimum solution, but it may propose alternative system designs to be compared to see which is best to meet the specific requirement. Simulation helps maintain better control over experimental conditions that would generally be possible when conducting experiments with real-world situations.

As stated by Pritsker, 1979, in its broadest sense, computer simulation is the process of designing a mathematical-logical model of a real system and experimenting with this model on a computer. The simulation process involves the following steps (Pritsker, 1986):

1. Process Formulation – definition of the problem to be studied including a statement of the problem-solving objective.
2. Model Building – abstraction of the system in mathematical logical relationships in accordance with the problem formulation.
3. Data Acquisition – identification, specification, and collection of data.
4. Model Translation – preparation of the model for computer processing.

5. Verification – process of establishing that the computer program executes as intended.
6. Validation – process of establishing that a desired accuracy or correspondence exists between the simulation model and the real system.
7. Strategic and Tactical Planning – process of establishing the experimental conditions for using the model.
8. Experimentation – execution of the simulation model to obtain output results.
9. Analysis of Results – process of analyzing the simulation outputs to draw references and make recommendations for problem resolution.
10. Implementation and Documentation – process of implementing decisions resulting from the simulation and documenting the model and its use.

Simulation involves systems and models of them.

A system is a facility or process, either actual or planned (Kelton, 2004). A few examples include:

- A manufacturing plant with people, machines, transport devices, conveyor belts, and storage space.
- A distribution network of plants, warehouses, and transportation links.
- An airport with departing passengers checking in, going through security, going to the departure gate, and boarding; departing flights contending for push-back tugs and runway slots; arriving flights contending for runways, gates, and arrival crew; arriving passengers moving to baggage claim and waiting for their bags; and the baggage-handling system dealing with delays, security issues, and equipment failure.

- A fast-food restaurant with different types of staff, customers, and equipment.

The following are the various parts to a simulation model (Kelton et al. 2004):

- Entities: dynamic objects in the simulation – they are usually created, moved around for a while, and then disposed of as they leave.
- Attributes: used to individualize entities, a common characteristic of all entities, but with a specific value that can differ from one entity to another.
- (Global) Variables: a piece of information that reflects some characteristic of the system, regardless of how many or what kinds of entities might be around.
- Resources: can represent a group of several individual servers that offer services to entities and are competed for.
- Queues: a place for the entity to wait.
- Statistical Accumulators: accumulators that help the user keep track of various variables.
- Events: something that happens at an instant of (simulated) time that might change attributes, variables, or statistical accumulators.
- Simulation Clock: current value of time in the simulation is simply held in a variable.

Simulation is not the only tool that can be used to study a model, but it is often the tool of choice over the last two or three decades. This is because it allows the system to become quite complex, but still represents the system faithfully. This is why simulation is the tool of choice for this research.

4.2 Arena, Version 5.0

As stated by the Arena software help menu, Arena is an easy-to-use, powerful tool that allows users to create and run experiments on models of their systems. By testing out ideas in this computer "laboratory" users can predict with confidence, and without disrupting their current business environment. Arena's unique template-based architecture provides the user with unparalleled ease of use, flexibility, and domain experience required in modeling any aspect of the business enterprise – from customer contact, manufacturing and business processes, to logistics and across the supply chain (Arena Help menu).

Any business environment, from manufacturing, to customer service, to health care, can benefit from simulation. Whether your analyzing an existing system or a new layout for a manufacturing facility, Arena allows users to just have to follow five easy steps (Arena Help Menu):

1. Create a basic model.
2. Refine the model.
3. Simulate the model.
4. Analyze simulation results.
5. Select the best alternative.

Arena is a Microsoft Office Compatible product. This means that its toolbars, menus, and accelerator keys are very similar to those used by Microsoft Office. Its menu structure follows standard Microsoft Office terminology and organization. Arena contains a toolbar that is very comparable to the ones in Microsoft Office. Most importantly, Arena has been designed to integrate in a natural fashion with Microsoft Office.

Arena represents process dynamics in a hierarchical flowchart and stores system information in data spreadsheets (Bapat 2003). To build models in Arena, users will be using modeling shapes, called modules, from the Basic Process panel (shown in Figures 4.1 and 4.2) to define the process. There are two types of modules on the panel:

1. Flowchart module – these are placed in the model window and connected to form a flowchart that describes the logic of the process (see Figure 4.1).
2. Data module – these are not placed in the model window, but are edited via a spreadsheet interface (see bottom 6 of Figure 4.2).

The model window contains two main regions. The main region, or model workspace, contains all of the model graphics, including the process flowchart, animation (if used), and other drawing elements. The other region displays model data, such as times, costs, and other parameters.

The Reports Panel (shown in Figure 4.3) lists various reports that are available to the user to display Arena simulation results. Reports may be generated at the end of the simulation run for final results, or at any time during the run to check the status of the system at a given point in the run. The user may generate as many reports as they would like. The eight reports provided by Arena include:

1. Category Overview
2. Entities
3. Processes
4. Queues

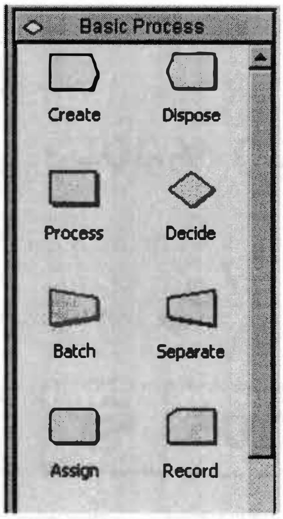


Figure 4.1 Basic Process Panel (Flowchart modules)

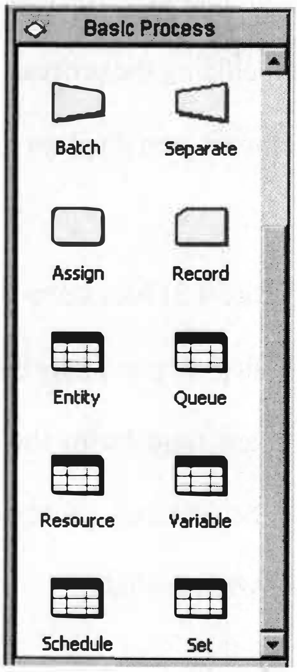
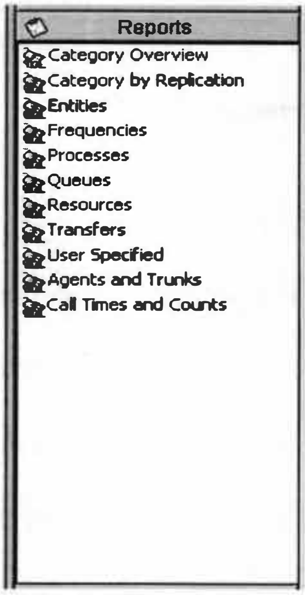


Figure 4.2 Basic Process Panel (Bottom 6 are Database modules)



Reports
Category Overview
Category by Replication
Entities
Frequencies
Processes
Queues
Resources
Transfers
User Specified
Agents and Trunks
Call Times and Counts

Figure 4.3 Arena Reports Panel

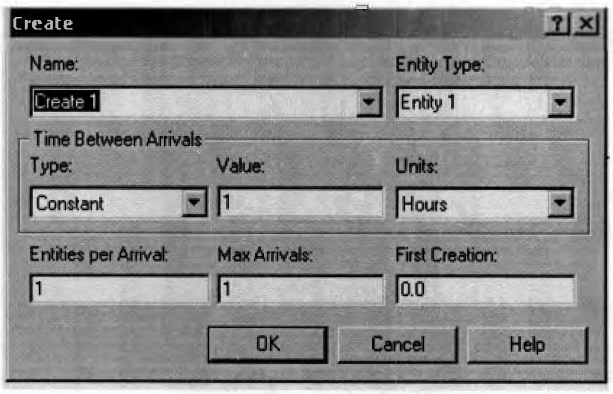
5. Resources
6. Transfers
7. User Specified
8. Frequencies

4.3 Simulation with Arena

The first step when using Arena is to create the model. This can be done by simply dragging Arena's modules – the shapes in the flowchart – into the model window and connecting them to define the process flow. The next step is to refine the model. This involves adding real-world data to the model by double clicking on the modules and adding the needed information to the Arena data forms. Next is the simulation of the model, which verifies the model properly reflects the actual system and identifies the system's bottlenecks. The last step involves the analyzing of the simulation results and the selection of the best model alternative.

4.3.1 The Create Flowchart Module

When creating a model in the Arena software, the first block that will be used is the Create module. The create module is the "birth" node for the arrival of entities to our model's boundary into the model from outside (Kelton et al. 2004). After having dragged the create module into the flowchart view, there will be the need to double click the block to open the dialog box (shown in figure 4.4) and enter the correct information for the each prompt. See appendix A under Create Module Prompts, if more explanation is needed on what each prompt is asking for.



Create

Name: Create 1

Entity Type: Entity 1

Time Between Arrivals

Type: Constant

Value: 1

Units: Hours

Entities per Arrival: 1

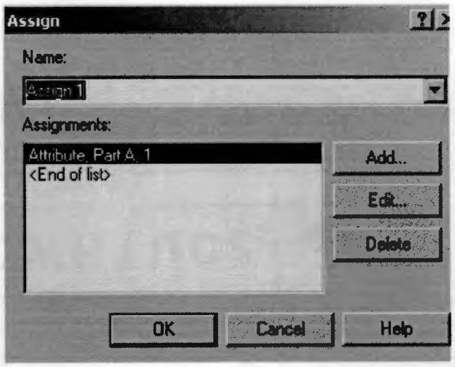
Max Arrivals: 1

First Creation: 0.0

Figure 4.4 The Create Property Dialog Box

4.3.2 The Assign Flowchart Module

The Assign module is used for assigning new values to variables, entity attributes, entity types, entity pictures, or other system variables. Multiple assignments can be made with a single Assign module. To use this module, the user needs to double click the block to open the dialog box (shown in figure 4.5) and enter the correct information for the each prompt. When the user is adding an attribute, entity type, or picture, there will be a need to click on the add button (from Figure 4.5) and the assign assignment box will appear (as shown in Figure 4.6). See appendix B under Assign Module Prompts, if more explanation is needed on what each prompt is asking for.

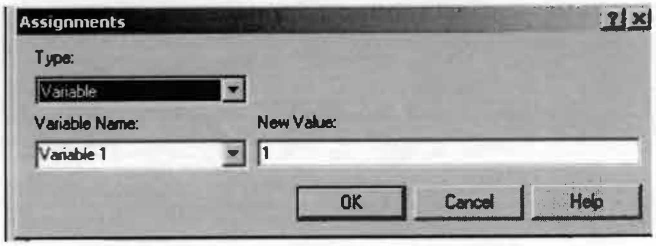


Assign

Name: Assign 1

Assignments:
Attribute, Part A, 1
<End of list>

Figure 4.5 The Assign Property Dialog Box



Assignments

Type: Variable

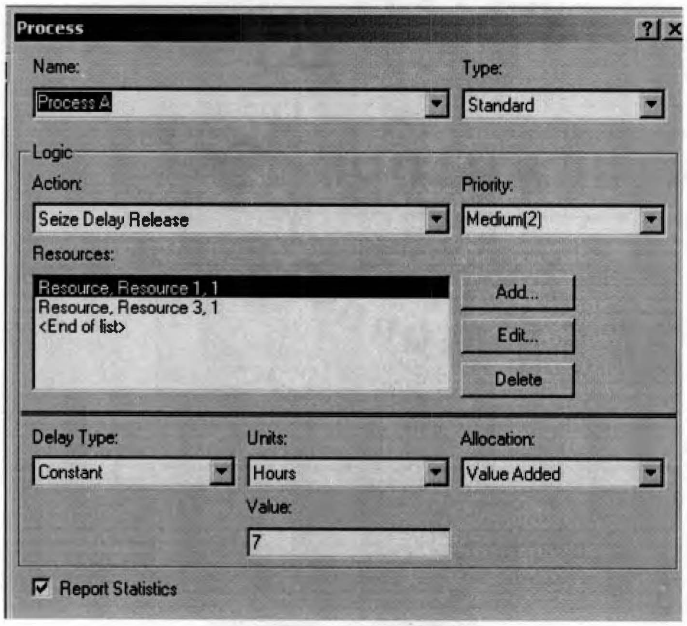
Variable Name: Variable 1

New Value: 1

Figure 4.6 The Assign Assignment Box

4.3.3 The Process Flowchart Module

The next block used when creating a simulation model is the process flowchart module. The process module can be used to represent the machine, its resource, its queue, and the entity delay time there. Again, to use this module, there will be the need to double click the block to open the dialog box (shown in figure 4.7) and enter the correct information for each prompt. See appendix C under Process Module Prompts, if more explanation is needed on what each prompt is asking for. When using Seize Delay Release, there will also be the need to double click the Add button and enter the proper information. Further explanation on how to use the Add button can be found in Section 4.3.4.



Process

Name: Process A

Type: Standard

Logic

Action: Seize Delay Release

Priority: Medium(2)

Resources:

Resource, Resource 1, 1
Resource, Resource 3, 1
<End of list>

Delay Type: Constant

Units: Hours

Allocation: Value Added

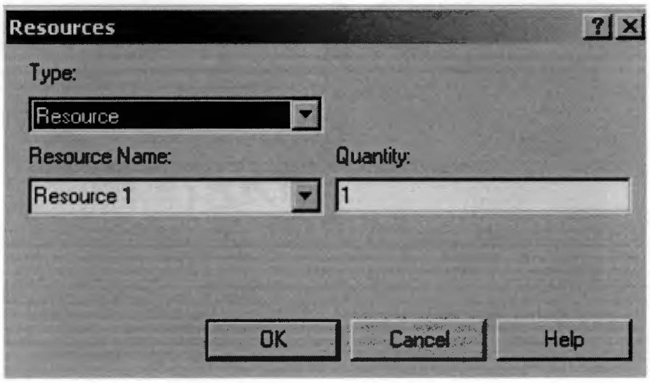
Value: 7

☑ Report Statistics

Figure 4.7 The Process Property Dialog Box

4.3.4 The Resource Data Module

This module is intended as the main processing method in the simulation. Options for seizing and releasing resource constraints are available. Additionally, there is the option to use a "sub model" and specify hierarchical user-defined logic. The process time is allocated to the entity and may be considered to be value added, non-value added, transfer, wait or other. The associated cost will be added to the appropriate category. This module pops up when the user clicks on the add button from the process dialog box. The user will then be asked to enter the correct information for each of the prompts (shown in Figure 4.8). See appendix D under Resource Module Prompts, if more explanation is needed on what each prompt is asking for.


Resources

Type: Resource

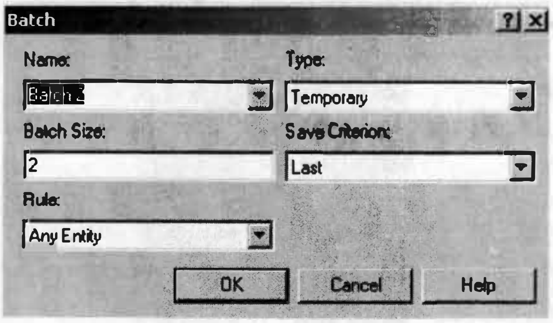
Resource Name: Resource 1

Quantity: 1

Figure 4.8 The Resource Dialog Box

4.3.5 The Batch Flowchart Module

This module is intended as the grouping mechanism within the simulation model. Batches of entities can be permanently or temporarily grouped. Temporary batches must later be split using the Separate module. Batches may be made with any specified number of entering entities or may be matched together based on an attribute. Entities arriving at the Batch module are placed in a queue until the required number of entities has accumulated. Once accumulated, a new representative entity is created. This module pops up when the user double clicks on the module. The user will then be asked to enter the correct information for each of the prompts (shown in Figure 4.9). See appendix E under Batch Module Prompts, if more explanation is needed on what each prompt is asking for.



Name: Batch 2

Type: Temporary

Batch Size: 2

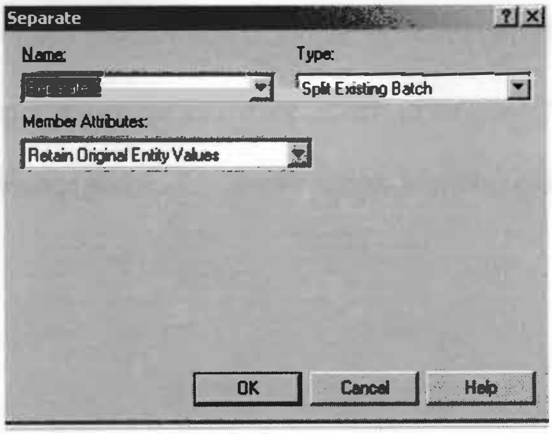
Save Criterion: Last

Rule: Any Entity

Figure 4.9 The Batch Property Dialog Box

4.3.6 The Separate Flowchart Module

This module can be used to either copy an incoming entity into multiple entities or to split a previously batched entity. Rules for allocating costs and times to the duplicate are specified. Rules for attribute assignment to member entities are specified as well. When splitting existing batches, the temporary representative entity that was formed is disposed of and the original entities that formed the group are recovered. The entities proceed sequentially from the module in the same order in which they originally were added to the batch. When duplicating entities, the specified number of copies is made and sent from the module. The original incoming entity also leaves the module. To use this module, there will be the need to double click the block to open the dialog box (shown in figure 4.10) and enter the correct information for each prompt. See appendix F under Separate Module Prompts, if more explanation is needed on what each prompt is asking for.



Separate

Name: Separate

Type: Split Existing Batch

Member Attributes: Retain Original Entity Values

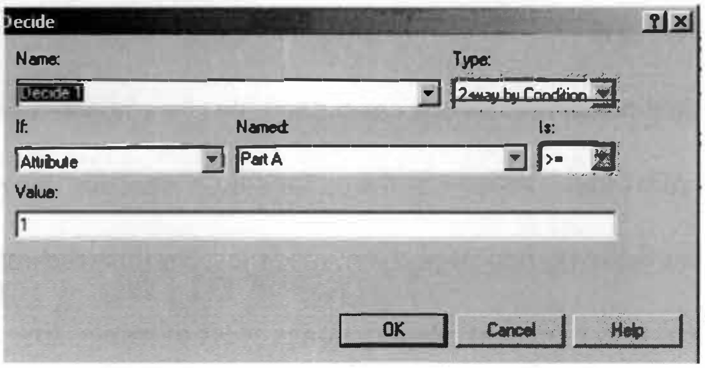
Figure 4.10 The Separate Property Dialog Box

4.3.7 The Decide Flowchart Module

The Decide module allows for decision-making processes in the system. It includes options to make decisions based on one or more conditions (e.g., if entity type is Gold Card) or based on one or more probabilities (e.g., 75% true; 25% false). Conditions can be based on attribute values (e.g., Priority), variable values (e.g., Number Denied), the entity type, or an expression (e.g., NQ (ProcessA.Queue)). To use this module, there will be the need to double click the block to open the dialog box (shown in figure 4.11) and enter the correct information for each prompt. See appendix G under Decide Module Prompts, if more explanation is needed on what each prompt is asking for.

4.3.8 The Dispose Flowchart Module

The Dispose module is the last of the modules used when creating your model in Arena. The Dispose model represents entities leaving the model boundaries (Kelton et al. 2004). By double clicking on its name, the dialog box will appear (shown in Figure 4.12). There will be a prompt to give the module a descriptive name and decide if output of entity statistics, which include things like average and maximum time in system of entities that go through this module and costing information of these entities, is wanted or not. For more explanation on these two prompts, please see appendix H under Dispose Module Prompts.



Decide

Name: Decide 1

Type: 2-way by Condition

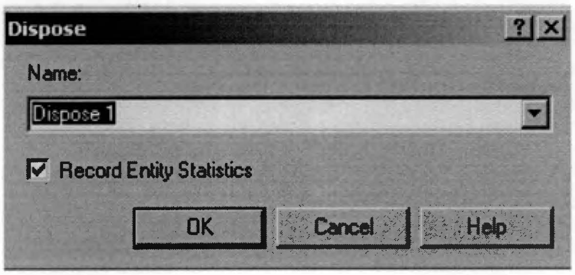
If: Attribute

Named: Part A

Is: >=

Value: 1

Figure 4.11 The Decide Property Dialog Box



Dispose

Name: Dispose 1

☑ Record Entity Statistics

Figure 4.12 The Dispose Property Dialog Box

4.4 Example Problem

A hypothetical problem has been created to help show how to use Arena, take advantage of most of its tools, and to test the methodology used for this research. Tables 4.1 and 4.2 represent the hypothetical problem made just for this research. This hypothetical problem is to represent a production process of some sorts. As is shown in Table 4.1, this problem (process) uses four resources. There is one unit available for resources 1 and 2 and two units available for resources 3 and 4. Table 4.2 shows that this problem (process) involves ten activities that need to be completed. This table also shows how long each activity is supposed to take and what resources are being used to complete each activity. The representation of the processes and their activity durations for the CRD method are shown in Table 4.3. Figures 4.13 – 4.24 represent the CPM and CRD networks for this hypothetical example problem.

Table 4.1 Resource Information

Resource No	Resource Type	Units Available
1	Manager	1
2	Assistant	1
3	Operator	2
4	Machine	2

Table 4.2 Sample Project Data

Activity No	Activity	Predecessor	Activity Duration	Resource Type Used
1	A		7	1 → 3
2	B		6	2
3	C	A,B	9	1 → 2 → 3
4	D	C	10	3 → 4
5	E	C	9	3 → 4
6	F	D	8	3 → 4
7	G	E	12	3 → 4
8	H	F	5	2 → 3
9	I	G,H	6	1 → 2 → 3
10	J	I	8	1

Table 4.3 CRD Sample Project Data

Process	Resource(s)	Duration
A	1 and 3	7
B	2	6
C1	1	3
C2	2	2
C3	3	4
D	3 and 4	10
E3	3	4
E4	4	4
F3	3	4
F4	4	4
G	3 and 4	12
H	2 and 3	5
I1	1	3
I23	2 and 3	3
J	1	8

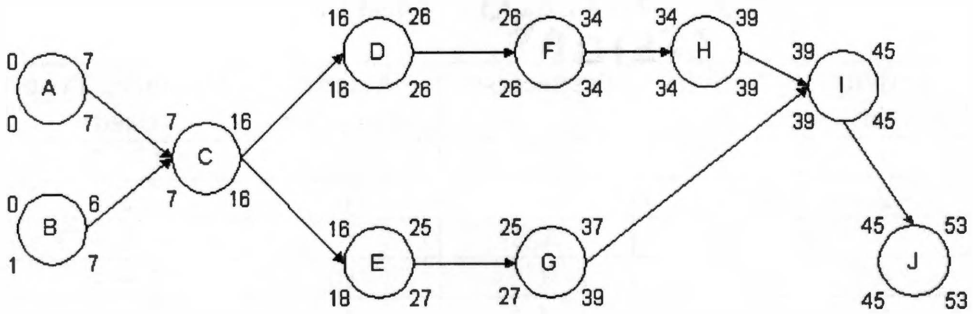


Figure 4.13 CPM Analysis

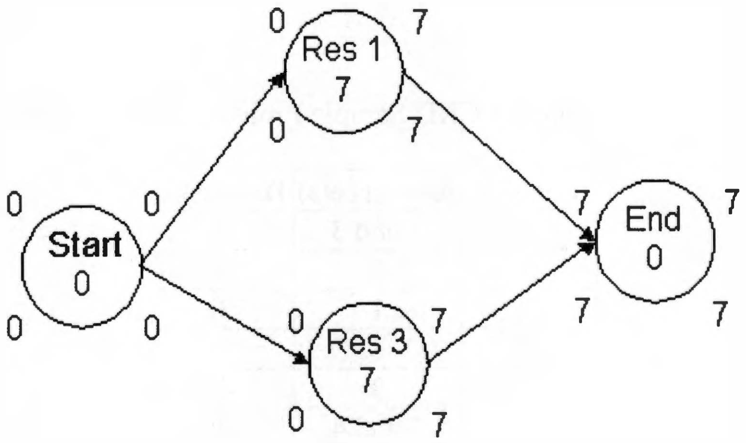


Figure 4.14 CRD Network for Activity A

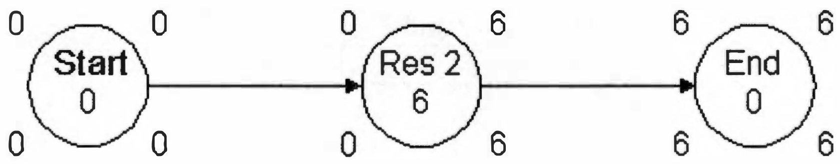


Figure 4.15 CRD Network for Activity B

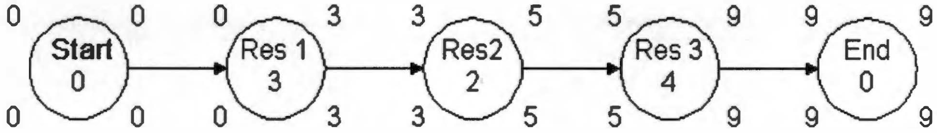


Figure 4.16 CRD Network for Activity C

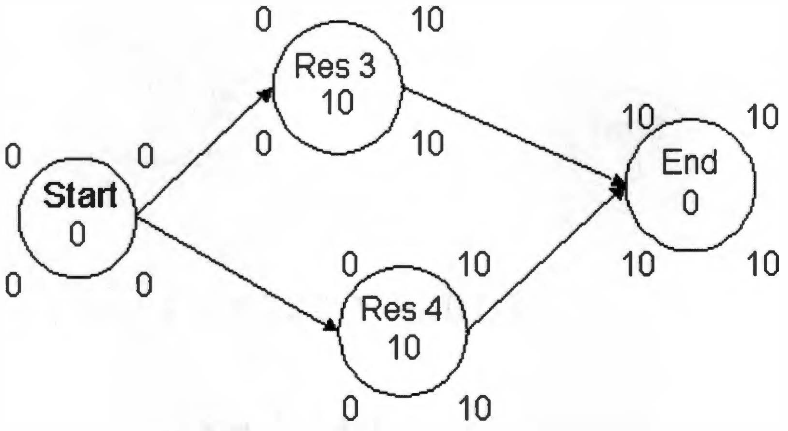


Figure 4.17 CRD Network for Activity D

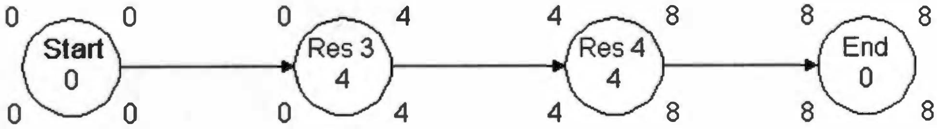


Figure 4.18 CRD Network for Activity E

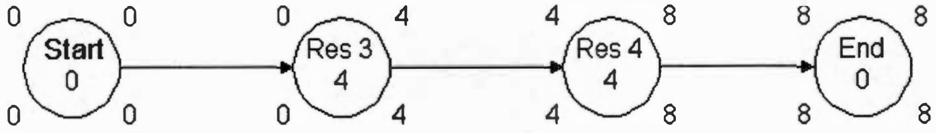


Figure 4.19 CRD Network for Activity F

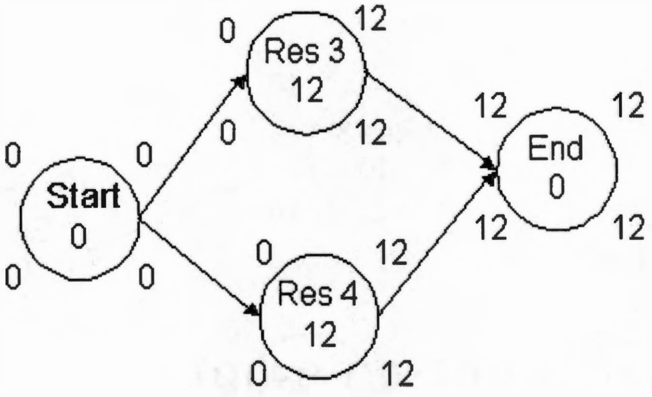


Figure 4.20 CRD Network for Activity G

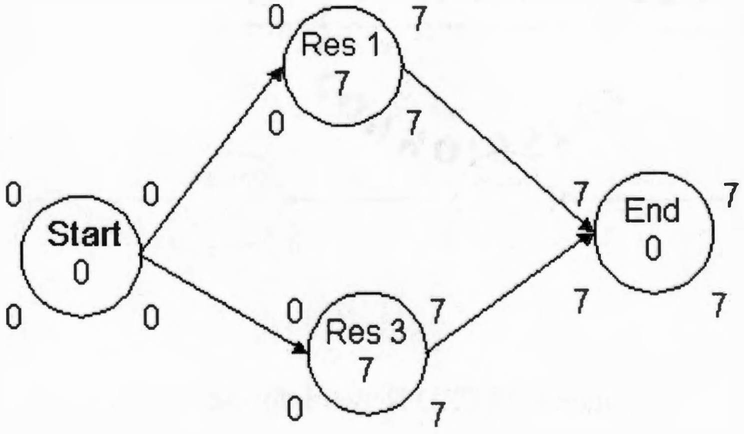


Figure 4.21 CRD Network for Activity H

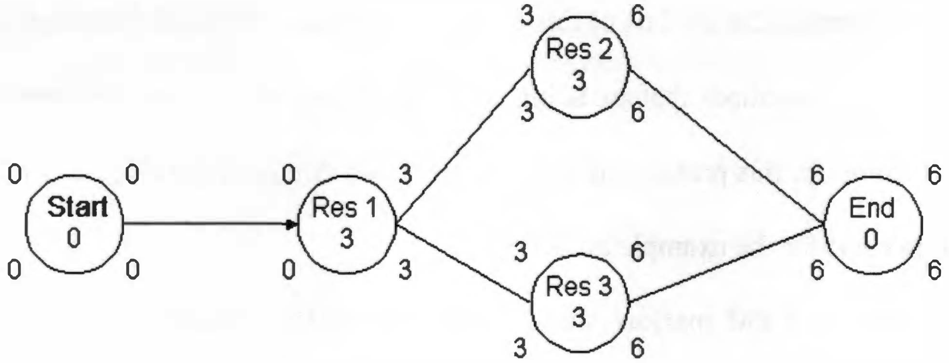


Figure 4.22 CRD Network for Activity I

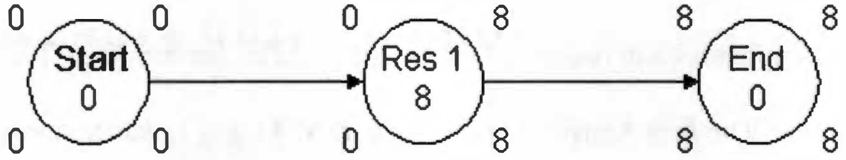


Figure 4.23 CRD Network for Activity J

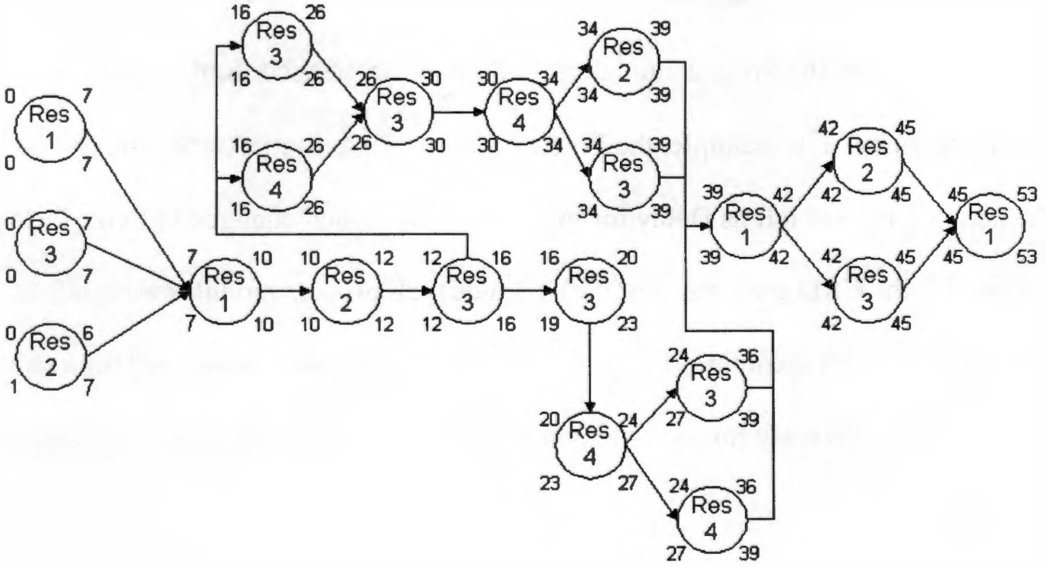


Figure 4.24 CRD Analysis

When creating the models of this problem in Arena, a different flowchart was created for the two methods that are being analyzed. Figure 4.25 shows the flowchart of the CPM method for this problem as it was entered into Arena. Figure 4.26 is the CRD method flowchart for the example problem.

For both the CPM analysis and the CRD analysis, two Create modules were needed. Create 1 was entered in the name box, entity type was left as it was, type was made as constant with a value of 1 and units of hours for both of the Create module dialog boxes. The entities per arrival was entered in as 1 with max arrivals of 1, and first creation at time 0.0 also for both of the create module dialog boxes.

This example problem required two Assign modules for both the CPM and CRD analysis in Arena. The first Assign module was named Assign 1, was assigned an attribute named Part A and was placed after the first Create module. The second assign module was named Assign 2, was assigned an attribute named Part B, and was placed after the second Create module.

Next were the Process modules. Each was named according to which activity it was to represent. For example, the Process module for activity A was named Process A. The action type was left as Delay for the CPM analysis and changed to Seize Delay Release for the CRD analysis. For both methods, all process modules were left as standard type, with medium priority, but the delay types, allocations, and time values were entered differently for each method depending on what type of problem was being prepared.

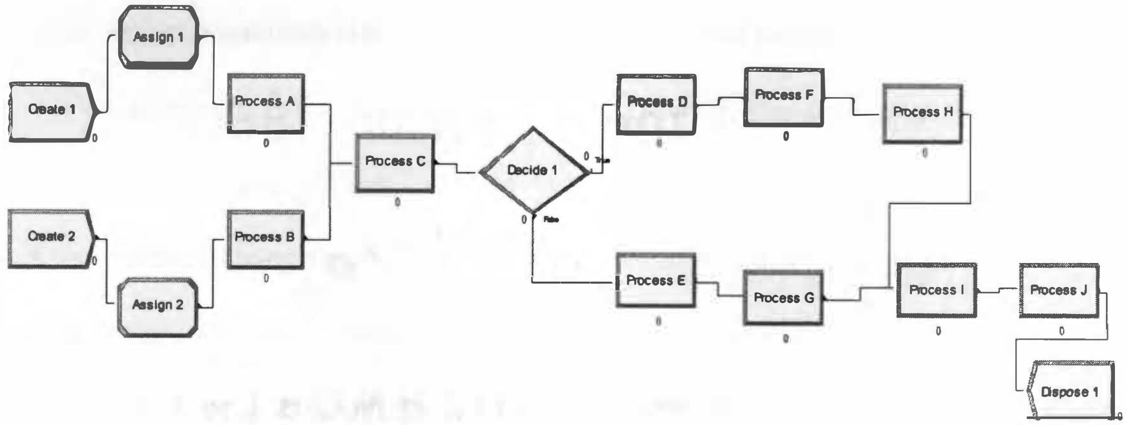


Figure 4.25 CPM Flowchart in Arena

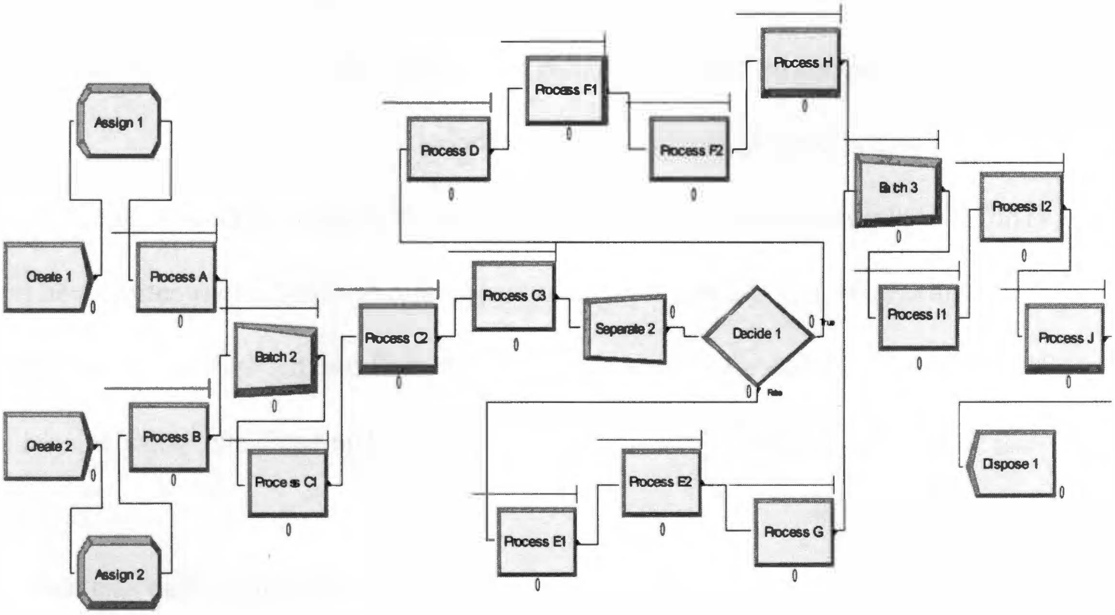


Figure 4.26 CRD Flowchart in Arena

The Resource module was not used for the CPM analysis of the example problem. It was, however, a key player in the CRD analysis. For the CRD simulation, Seize Delay Release was chosen as the action from the Process module. This gives the option of adding a resource, multiple resources, or sets that are to be used for this process. Resources were added accordingly as to which process they are assigned to, which can be figured using Table 4.2. One example from the mentioned table is both resources 1 and 3 were added to Process A for the CRD method. Adding both these resources in the Process A module assumes that both these resources are being used on this process at the same time. If the resources for a process were not being used at the same time, a separate process had to be created to show this. An example of this is Process C. Process C requires resources 1, 2, and 3, but none of the resources can be utilized at the same time. This problem was handled by creating three separate processes (Process C1, Process C2, and Process C3) and allocating one resource per process.

The CRD simulation approach required two batch modules. The first Batch module was named Batch 2 and was labeled as temporary, so that the two entities can be split after Process C. The second Batch module was named Batch 3 and was labeled as permanent, so that the entities to leave the process as one. The batch size was declared as two for both the modules.

The CRD simulation approach also required a Separate module. This was used to separate the two entities after Process C, that had been batched together beforehand. This module was named Separate 2. Its type was labeled as split two entities. Its member attributes were to retain original entity values.

The Decide module was the same for both the CPM and CRD analysis. The name prompt was left as it was. The type was changed to 2 way by condition. The condition was set at if the attribute is named Part A with a value of 1, it was to go to Process D, otherwise (being Part B) it was to go to Process E.

For both the CPM and CRD analysis of the sample problem, the Dispose module prompts were left as they were.

4.5 Simulation Results

The next step after creating the flowchart is to run the simulation and retrieve the results. Running two different types of simulations per method did this. The first approach used the deterministic times for processes, which assumes the process times are known with great certainty. The second approach used a probability for the process times, which assumes the times are not known with certainty. The following results were acquired from ten runs for each approach.

4.5.1 Deterministic Results

Table 4.4, shows the results of the Arena simulation for the CPM method, while Table 4.5, 4.6, and 4.7 show the simulation results from Arena for the CRD method. The process times for this method were set as constant, because they are assumed known with great certainty. They were then imputed as the expected times of each activity.

Table 4.4 CPM Deterministic Arena Results

Activity	Total Process Time
Process A	7 hours
Process B	6 hours
Process C	9 hours
Process D	10 hours
Process E	8 hours
Process F	8 hours
Process G	12 hours
Process H	5 hours
Process I	6 hours
Process J	8 hours
Total	53 hours

Table 4.5 CRD Resource Deterministic Arena Results

Resource	Utilization	Times Used
1	.39623	4
2	.30189	4
3	.46226	8
4	.28302	4

Table 4.6 CRD Deterministic Batch Results

Batch	Queue Waiting Time	Queue Number In
2	0.5	0.01887
3	1.5	0.05660

Table 4.7 CRD Deterministic Arena Results

Activity	Total Process Time
Process A	7 hours
Process B	6 hours
Process C1	3 hours
Process C2	2 hours
Process C3	4 hours
Process D	10 hours
Process E1	4 hours
Process E2	4 hours
Process F1	4 hours
Process F2	4 hours
Process G	12 hours
Process H	5 hours
Process I1	3 hours
Process I2	3 hours
Process J	8 hours
Total	53 hours

4.5.2 Probabilistic Results

Table 4.8, shows the probabilistic results of the Arena simulation for the CPM method, while Table 4.9, 4.10, and 4.11 show the probabilistic simulation results from Arena for the CRD method.

These results were obtained using a normal distribution with a mean of the expected process times (process times used in deterministic portion of the problem) and a standard deviation of 0.5. Many other distributions could have been used and tested, but with the limited constraints of this research, the issue could not be researched further. To change the distribution type that is being used, the user clicks on the Delay Type from the Process module and chooses the type of distribution needed.

Table 4.8 CPM Probabilistic Arena Results

Activity	Min Process Time	Max Process Time
Process A	6.0703	7.4746
Process B	5.6416	7.0441
Process C	8.2158	9.8443
Process D	8.6278	11.149
Process E	7.7306	8.3206
Process F	7.1313	8.7934
Process G	11.067	13.604
Process H	4.4405	5.4471
Process I	4.8704	6.7641
Process J	7.4518	9.0867
Total	50.5214	54.0070

Table 4.9 CRD Resource Probability Arena Results

Resource	Min Utilization	Max Utilization	Times Used
1	.35858	.42523	4
2	.28190	.34219	4
3	.44709	.46972	8
4	.26102	.29392	4

Table 4.10 CRD Probabilistic Batch Results

Batch	Average Queue Wait Time	Average Queue Number In
2	0.376621	0.014034
3	1.627689	0.060472

Table 4.11 CRD Probabilistic Arena Results

Activity	Min Process Time	Max Process Time
Process A	6.0703	7.5071
Process B	4.6777	7.2759
Process C1	2.6416	4.0441
Process C2	1.2158	2.6444
Process C3	3.3075	4.8443
Process D	8.6278	10.801
Process E1	3.3243	4.5447
Process E2	3.0670	5.6040
Process F1	3.6702	4.7934
Process F2	3.4405	4.7641
Process G	11.131	12.525
Process H	3.8704	5.4583
Process I1	2.3848	3.9017
Process I2	2.3973	4.0867
Process J	6.8776	9.2861
Total	51.6236	55.0999

CHAPTER 5

CONCLUSIONS

5.1 Conclusions

The Arena output results for the deterministic parts of the example were compared to the results of the done by hand portion of the hypothetical problem. The values obtained were very consistent and matched perfectly. Thus, Arena can be used as an effective and accurate management tool when combined with the CPM and CRD methods.

When using simulation, specifically Arena, it would be very easy for the user to enter the processes and the information needed by Arena and run the model. Trying to find the results for either method by hand for a very large project would be very tedious and would take a very long time. The deterministic results entail that Arena is very consistent and matches the method done by hand, and could be very beneficial for large projects.

Arena may also be useful when the user does not know the exact process times of the activities. When entering the process time into the Process Dialog Box, there is the option of choosing a distribution for the process time. As long as the user has an idea of what type of distribution they want to use, a mean process time for each activity in that distribution, and the standard deviation they want the process to follow, Arena can be used to give the user minimum, average, and maximum process times for each individual process.

Arena is both easy and a user-friendly simulation software, thus providing a very simple and important tool in the area of project management. In summary, the research reported here has made a significant contribution in enhancing the CPM method and applying the CRD method to a new application, especially for large projects.

The CPM and CRD methods are both very useful methods for project scheduling. The main difference is that CPM concentrates on the activities and their process times, while CRD emphasizes the resources and the amount of time each resource is being used. This concludes that CRD is the better method when concerned with resources and the scheduling of them.

5.2 Future Research

The example problem used in this research is too small and simple, making it unrealistic. For more realistic results, a more complicated model may be required. This could be done by using a real world example and running it in Arena, and comparing its results to the results found by implementing the problem in the real world.

Not included in the scope of this research is the comparison of Arena with other simulation softwares. Arena is not, by far, the only simulation software available. There are many other simulation software packages available. Some examples include ProcessModel, FlexSim, SimCad and many more. In the future, a comparison of Arena with these other softwares, using the same or similar approach of this research may be a very beneficial project or area of research.

REFERENCES

REFERENCES

Ahuja, H.N., *Construction Performance Control by Networks*, Wiley, New York, 1976.

Badiru, Adedeji B., "A Project Scheduling Heuristic under Stochastic Time and Resource Constraints (STARC)," doctoral dissertation, University of Central Florida, Orlando, Florida, 1984.

Badiru, Adedeji B., *Project Management in Manufacturing and High Technology Operations,* New York: John Wiley & Sons, 1988c.

Badiru, Adedeji B., "Critical Resource Diagram: A New Tool for Resource Management," *Industrial Engineering,* Vol. 24, No. 10, October 1992, pp. 58-65.

Badiru, Adedeji B., "Activity-Resource Assignments Using Critical Resource Diagramming," *Project Management Journal,* Vol. 24, No. 3, September 1993, pp. 15-21.

Badiru, Adedeji B., and S.P. Pulat, *Comprehensive Project Management: Integrating Optimization Models, Management Principles and Computers,* New Jersey: Prentice Hall, 1995.

Balas, E., "Project Scheduling with Resource Constraints," *Applications of Mathematical Techniques,* Carnegie-Mellon University, Pittsburgh, 1970.

Bandelloni, M., M. Tucci, and R. Rinaldi, "Optimal Resource Leveling using Non-Serial Dynamic Programming," *European Journal of Operational Research,* Vol. 78, 1994, pp. 162-177.

Bapat, V. and D.T. Sturrock, "The Arena Product Family: Enterprise Modeling Solutions," *Proceedings of the 2003 Winter Simulation Conference,* 2003, pp. 210-217.

Bedworth, D.D., *Industrial Systems: Planning, Analysis, and Control,* New York: Ronald Press, 1973.

Bedworth, D.D., and J.E. Bailey, *Integrated Production Control Systems: Management, Analysis, and Design,* New York, John Wiley, 1982.

Beimborn, E.A. and W.A. Gavey, "The Blob Chart," *Industrial Engineering,* Vol. 4, No. 12, December 1972, pp. 17-19.

Bell, C.E. and K. Park, "Solving Resource-Constrained Project Scheduling Problems by A^* Search, *Naval Research Logistics Quarterly,* Vol. 37, No. 1, 1990, pp. 61-84.

Biegel, J.E. and M. Halim, "A Graphical Method for Project Resource Allocation," *1975 Proceedings of the AIIE Systems Engineering Conference,* Norcross, Georgia, 1975, pp. 333-336.

Brooks, G.H. and J.B. White, "An Algorithm for Finding Optimal or Near Optimal Solutions to the Production Scheduling Problem," *Journal of Industrial Engineering,* Vol. 16, January-February 1965, pp. 34-40.

Burgess, A.R., and J.B. Killebrew, "A Network Interpretation and a Directed Subdual Algorithm for Critical Path Scheduling," *Journal of Industrial Engineering*, Vol. 13, No. 2, March-April 1962.

Chan, F.T.S., "Resource Management in Project Scheduling Through Simulation," *International Journal of Computer Applications in Technology*, Vol. 10, Nos. 1-2, 1997, pp. 4-5.

Charnes, A. and W.W. Cooper, "A Network Interpretation and a Directed Subdual Algorithm for Critical Path Scheduling," *Journal of Industrial Engineering,* Vol. 13, No. 4, 1962, pp. 213-218.

Christofides, N.R., Alvarez-Valdes, and J.M Tamarit, "Project Scheduling with Resource Constraints: A Branch and Bound Approach," *European Journal of Operations Research,* Vol. 29, 1987, pp. 262-273.

Collins, N. and C.M. Watson, "Introduction to ARENA," *Proceedings of the 1993 Winter Simulation Conference,* 1993, pp. 205-212.

Davis, E.W., "An Exact Algorithm for the Multiple Constrained Resource Project Scheduling Problem," unpublished Ph.D. dissertation, Yale University, 1969.

Davis, E.W., "Project Scheduling under Resource Constraints: Historical Review and Categorization of Procedures," *AIIE Transactions*, Vol. 5, No. 4, December 1973, pp. 297-313.

Davis E.W. and G.E. Heidorn, "Optimal Project Scheduling under Multiple Resource Constraints," *Management Science,* Vol. 17, No. 12, August 1971, pp. B803-B816.

Davis, E. W., and J. H. Patterson, "A Comparison of Heuristic and Optimal Solutions in Resource Constrained Project Scheduling," *Management Science,* Vol. 21, No. 8, April 1975, pp. 944-955.

Demeulemeester, E., "Minimizing Resource Availability Costs in Time-Limited Project Networks," *Management Science*, Vol. 41, 1995, pp. 1590-1598.

Easa, S.M., "Resource Leveling in Construction by Optimization," *Journal of Construction Engineering and Management*, Vol. 115, 1989, pp. 302-316.

Elaysed, E.A., "Algorithms for Project Scheduling with Resource Constraints," *International Journal of Production Research,* Vol. 20, No. 1, 1982, pp. 95-103.

Fisher, M.L., "Optimal Solution of Scheduling Problems Using Lagrange Multipliers, Part I," *Operations Research,* Vol. 21, No. 5, September-October 1973, pp. 1114-1127.

Fulkerson, D.R., "A Network Flow Computation for Project Cost Curves," *Management Science,* Vol. 7, No. 2, January 1961, pp. 167-179.

Gorenstein, S., "An Algorithm for Project Scheduling with Resource Constraints," *Operations Research,* Vol. 20, No. 4, July-August 1972, pp. 835-850.

Ingalls, R.G., "Introduction to Simulation," *Proceedings of the 2002 Winter Simulation Conference,* 2002, pp. 7-16.

Jayaraman, H., "An Improved Activity Prioritizing Heuristic in Stochastic PERT Networks," MS Thesis, University of Oklahoma, 1995.

Johnson, R.V., "Resource Constrained Scheduling Capabilities of Commercial Project Management Software," *Project Management Journal,* Vol. 22, No. 4, December 1992, pp. 39-43.

Kelton, W.D., R.P. Sadowski, and D.T. Sturrock, *Simulation With Arena*, New York: McGraw-Hill Companies, Inc., 2004

Kerzner, Harold, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling,* Ohio: John Wiley & Sons, Inc., 2003.

Levy, F.K, G.L. Thompson, and J.D. Wiest, "Multi-Ship, Multi-Shop, Workload Smoothing Program," *Naval Research Logistics Quarterly,* Vol. 9, No. 1, March 1962.

Moder, J.J. and C.R. Phillips, *Project Management with CPM and PERT,* New York: Van Nostrand Reinhold Company, 1970.

Neumann, K. and J. Zimmermann, "Procedures for Resources Leveling and Net Present Value Problems in Project Scheduling with General Temporal and Resource Constraints," *European Journal of Operational Research*, Vol. 127, 2000, pp. 425-443.

Patterson, J.H. and W.D. Huber, "A Horizon Varying Zero-One Approach to Project Scheduling," *Management Science,* Vol. 20, No. 6, February 1974. pp. 990-998.

Patterson, J.H. and G. Roth, "Scheduling a Project under Multiple Resource Constraints: A Zero-One Programming Approach," *AIIE Transactions,* Vol. 8, No. 3, December 1976, pp. 449-456.

Pritsker, A.A.B, "Compilation of Definitions of Simulation," *SIMULATION*, Vol. 33, 1979, pp. 61-63.

Pritsker, A.A.B., *Introduction to Simulation and SLAM II.* New York: John Wiley & Sons, Inc., 1986.

Rockwell Software, *Arena, version 5.0*, McGraw Hill Companies, Inc., Sewickley, PA, 1993-2003.

Savin, D., S. Alkass, and P. Fazio, "Construction Resource Leveling Using Neural Networks," *Canadian Journal of Civil Engineering*, Vol. 23, 1996, pp. 917-925.

Savin, D., S. Alkass, and P. Fazio, "A Procedure for Calculating the Weight-matrix of a Neural Network For Resource Leveling," *Advances of Engineering Software*, Vol. 28, 1997, pp. 227-283.

Schrage, L., "Solving Resouce-Constrained Network Problems by Implicit Enumeration-Non-Preemptive Case," *Operations Research,* Vol. 18, No. 2, March-April 1970, pp. 225-235.

Shannon, R.E., *Systems Simulation – The Art and Science,* Prentice Hall, 1975.

Simmons, L.F., "Project Management – Critical Path Method (CPM) and PERT Simulated with ProcessModel," *Proceedings of the 2002 Winter Simulation Conference*, 2002, pp. 1789-1788.

Stinson, J.P, E.W. Davis, and B.M. Khumawala, "Multiple Resource-Constrained Scheduling Using Branch and Bound," *AIIE Transactions,* Vol. 10, No. 3, September 1978, pp. 252-259.

Talbot, F.B., "An Integer Programming Algorithm for the Resource-Constrained Project Scheduling Problem," unpublished Ph.D. Dissertation, Pennsylvania State University, 1976.

Talbot, F.B., "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case," *Management Science,* Vol. 28, No. 10, October 1982, pp. 1197-1210.

Talbot, F.B. and J.H. Patterson, “An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems,” *Management Science,* Vol. 24, No. 11, July 1978, pp. 1163-1174.

Whitehouse, G.E., and J.R. Brown, “GENRES: An Extension of Brooks Algorithm for Project Scheduling with Resource Constraints,” *Computers and Industrial Engineering,* Vol. 3, No. 4, December 1979, pp. 261-268.

Wiest, J.D., “A Heuristic Model for Scheduling Large Projects with Limited Resources,” *Management Science,* Vol. 13, February 1967, pp. B359-B377.

Wiest, J.D., and F.K. Levy, *A Management Guide to PERT/CPM,* New Jersey: Prentice Hall, Inc., 1969.

Younis, M.A., and B. Saad, “Optimal Resource Leveling of Multi-Resource Projects,” *Computers and Industrial Engineering,* Vol. 31, 1996, pp. 1-4.

APPENDICES

Appendix A: Create Module Prompts

Prompt	Definition	Valid Entry	Default
Name	Unique module identifier. This name is displayed on the module shape.	Symbol Name [All modules]	<module name and instance number>
Entity Type	Name of the entity type to be generated.	Symbol Name [Entity Name]	Entity 1
Type	Type of arrival stream to be generated. Types include: Random (uses an Exponential distribution, user specifies mean), Schedule (uses an Exponential distribution, mean determined from the specified Schedule module), Constant (user specifies constant value, e.g., 100), or Expression (pull down list of various distributions).	Random (Expo), Schedule, Constant, Expression	Random
Value	Determines the mean of the exponential distribution (if Random is used) or the constant value (if Constant is used) for the time between arrivals. Applies only when Type is Random or Constant.	Real	1
Schedule Name	Identifies the name of the schedule to be used. The schedule defines the arrival pattern for entities arriving to the system. Applies only when Type is Schedule.	Symbol Name [Schedules]	Schedule 1
Expression	Any distribution or value specifying the time between arrivals. Applies only when Type is Expression.	Expression (Distributions)	1
Units	Time units used for interarrival and first creation times. Does not apply when Type is Schedule.	Seconds, Minutes, Hours, Days	Hours
Entities per Arrival	Number of entities that will enter the system at a given time with each arrival.	Expression	1

Max Arrivals	Maximum number of entities that this module will generate. When this value is reached, the creation of new entities by this module ceases.	Expression	Infinite
First Creation	Starting time for the first entity to arrive into the system. Does not apply when Type is Schedule	Expression	0.0

Appendix B: Assign Module Prompts

Prompt	Definition	Valid Entry	Default
Name	Unique module identifier. This name is displayed on the module shape.	Symbol Name [All Modules]	<module name and instance number>
Assignments	Specifies the one or more assignments that will be made when an entity executes the module.		
Type	Type of assignment to be made. Other can include system variables, such as resource capacity or simulation end time.	Variable, Attribute, Entity Type, Entity Picture,Other	Variable
Variable Name	Name of the variable that will be assigned a new value when an entity enters the module. Applies only when Type is Variable.	Symbol Name [Variables]	Variable 1
Attribute Name	Name of the entity attribute that will be assigned a new value when the entity enters the module. Applies only when Type is Attribute.	Symbol Name [Attributes]	Attribute 1
Entity Type	New entity type that will be assigned to the entity when the entity enters the module. Applies only when Type is Entity Type.	Symbol Name [Entity Types]	Entity 1
Entity Picture	New entity picture that will be assigned to the entity when the entity enters the module. Applies only when Type is Entity Picture.	Symbol Name [Entity Pictures]	Picture.Report
Other	Identifies the special system variable that will be assigned a new value when an entity enters the module. Applies only when Type is Other.	Expression	J
New Value	Assignment value of the attribute, variable, or other system variable. Does not apply when Type is Entity Type or Entity Picture	Expression	1

Appendix C: Process Module Prompts

Prompt	Definition	Valid Entry	Default
Name	Unique module identifier. This name is displayed on the module shape.	Symbol Name [All Modules]	<module name and instance number>
Type	Method of specifying logic within the module. Standard processing signifies that all logic will be stored within the Process module and defined by a particular Action. Submodel indicates that the logic will be hierarchically defined in a "submodel" that can include any number of logic modules.	Standard, Submodel	Standard
Action	Type of processing that will occur within the module. Delay simply indicates that a process delay will be incurred with no resource constraints. Seize Delay indicates that a resource(s) will be allocated in this module and a delay will occur, but that resource release will occur at a later time. Seize Delay Release indicates that a resource(s) will be allocated followed by a process delay and then the allocated resource(s) will be released. Delay Release indicates that a resource(s) has previously been allocated and that the entity will simply delay and release the specified resource(s). Applies only when Type is Standard.	Delay, Seize Delay, Seize Delay Release, Delay Release	Delay
Priority	Priority value of the entity waiting at this module for the specified resource(s). Used when one or more entities from other modules are waiting for the same resource(s). Does not apply when Action is Delay or Delay Release, or when Type is Submodel.	High (1), Medium (2), Low (3), Other Expression	Medium (2)
Delay Type	Type of distribution or method of specifying the delay parameters. Constant and Expression require single values, while Normal, Uniform and Triangular require several parameters.	Constant, Normal, Triangular, Uniform, Expression	Triangular
Units	Time units for delay parameters.	Seconds, Minutes, Hours, Days	Hours

Allocation	Determines how the processing time and process costs will be allocated to the entity . The process may be considered to be value added, non-value added, transfer, wait or other and the associated cost will be added to the appropriate category for the entity and process.	Value Added, Non-Value Added, Transfer, Other, Wait	Value Added
Minimum	Parameter field for specifying the minimum value for either a Uniform or Triangular distribution.	Expression	.5
Value	Parameter field for specifying the mean for a Normal distribution, the value for a Constant time delay, or the mode for a Triangular distribution.	Expression	1
Maximum	Parameter field for specifying the maximum value for either a Uniform or Triangular distribution.	Expression	1.5
Std Dev	Parameter field for specifying the standard deviation for a Normal distribution.	Expression	.2
Expression	Parameter field for specifying an expression whose value is evaluated and used for the processing time delay.	Expression	1
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this process.	Checked, Unchecked	<Checked>

Appendix D: Resource Dialog Box Prompts

Prompt	Definition	Valid Entry	Default
Resources	Lists the resources or resource sets used for entity processing. Does not apply when Action is Delay, or when Type is Submodel.		
Type	Specification of a particular resource, or selecting from a pool of resources (i.e., a resource set).	Resource, Set	Resource
Resource Name	Name of the resource that will be seized and/or released. Applies only when Type is Resource.	Symbol Name [Resources]	Resource 1
Quantity	Number of resources of a given name or from a given set that will be seized/released. For sets, this value specifies only the number of a selected resource that will be seized/released (based on the resource's capacity), not the number of members of a set to be seized/released.	Expression Truncated to Integer	1

Appendix E: Batch Module Prompts

Prompt	Definition	Valid Entry	Default
Name	Unique module identifier. This name is displayed on the module shape.	Symbol Name [All Modules]	<module name and instance number>
Type	Method of batching entities together.	Temporary, Permanent	Permanent
Batch Size	Number of entities to be batched.	Expression Truncated to Integer	2
Save Criterion	Method for assigning representative entity's user defined attribute values.	First, Last, Sum, Product	Last
Rule	Determines how incoming entities will be batched. Any Entity will take the first "Batch Size" number of entities and put them together. By Attribute signifies that the values of the specified attribute must match for entities to be grouped. For example, if Attribute Name is Color, all entities must have the same Color value to be grouped. Otherwise, they will wait at the module for additional incoming entities.	Any Entity, By Attribute	Any Entity
Attribute Name	Name of the attribute whose value must match the value of the other incoming entities in order for a group to be made. Applies only when Rule is By Attribute.	Symbol Name [Attributes]	Attribute 1

Appendix F: Separate Module Prompts

Prompt	Definition	Valid Entry	Default
Name	Unique module identifier. This name is displayed on the module shape.	Symbol Name [All Modules]	<module name and instance number>
Type	Method of separating the incoming entity. Duplicate Original will simply take the original entity and make some number of duplicates. Split Existing Batch requires that the incoming entity be a temporarily batched entity using the Batch module. The original entities from the batch will be split.	Duplicate Original, Split Existing Batch	Duplicate Original
Percent Cost to Duplicates	Allocation of costs and times of the incoming entity to the outgoing duplicates. This value is specified as a percentage of the original entity's costs and times (between 0-100). The percentage specified will be split evenly between the duplicates, while the original entity will retain any remaining cost/time percentage. Applies only when Type is Duplicate Original.	Expression	50
# of Duplicates	Number of outgoing entities that will leave the module, in addition to the original incoming entity. Applies only when Type is Duplicate Original	Expression	1
Member Attributes	Method of determining how to assign the representative entity attribute values to the original entities. These options relate to six of the special purpose attributes (Entity.Type, Entity.Picture, Entity.Station, Entity.Sequence, Entity.Jobstep, and Entity.HoldCostRate) and all user defined attributes. Applies only when Type is Split Existing Batch.	Retain Original Entity Values, Take All Representative Values, Take Specific Representative Values	Retain Original Entity Values
Attribute Name	Name of representative entity attribute(s) that is assigned to original entities of the group. Applies only when Member Attributes is Take Specific Representative Values.	Symbol Name [Attributes]	Attribute 1

Appendix G: Decide Module Prompts

Prompt	Definition	Valid Entry	Default
Name	Unique module identifier. This name is displayed on the module shape.	Symbol Name [All Modules]	<module name and instance number>
Type	Indicates whether the decision is based on a condition or by chance or percentage. The type can be specified as either 2-way or N-way. 2-way allows for one condition or probability N-way allows for any number of conditions or probabilities to be specified as well as an "else" exit.	2-way by Condition, 2-way by Chance. N-way by Condition, N-way by Chance	2-way by Chance
Conditions	Defines one or more conditions used to direct entities to different modules Applies only when Type is N-way by Condition.		
Percentages	Defines one or more percentages used to direct entities to different modules. Applies only when Type is N-way by Chance.		
Percent True	Value that will be checked to determine the percentage of entities sent out a given true exit.	Expression (Various percentage alternatives)	50
If	Type of conditions that are available for evaluation	Variable, Type, Attribute, Entity Expression	Entity Type
Named	Specifies either the name of the variable, attribute, or entity type that will be evaluated when an entity enters the module. Does not apply when Type is Expression.	Symbol Name [Variables, Attributes, Entity Types]	Variable 1, Attribute 1, Entity 1
Is	Evaluator for the condition. Applies only to Attribute and Variable conditions.	>=, >, ==, <>, <, <=	>=
Value	Expression that will be either compared to an attribute or variable, or that will be evaluated as a single expression to determine if it is true or false. Does not apply to Entity		

Appendix H: Dispose Module Prompts

Prompt	Definition	Valid Entry	Default
Name	Unique module identifier. This name is displayed on the module shape.	Symbol Name [All Modules]	<module name and instance number>
Record Entity Statistics	Determines whether or not the incoming entity's statistics will be recorded. Statistics include value added time, non-value added time, wait time, transfer time, other time, total time, value added cost, non-value added cost, wait cost, transfer cost, other cost, and total cost.	Checked, Unchecked	<Checked>

VITA

Jamie Ehresman Gunter was born on February 5, 1980 in Lansing, Michigan as Jamie Ann Ehresman. She grew up in Mason, Michigan and graduated from Mason High School in 1998. She then moved to Tennessee where she attended the University of Tennessee, Knoxville, and obtained a Bachelor of Science in Industrial Engineering in December of 2002. In January 2003, she entered the master's program at the University of Tennessee, Knoxville, and worked as a graduate assistant for Dr. Adedeji Badiru. She received her Master's degree in Industrial Engineering in December 2004. Mrs. Gunter is going to start work at UPS on August 16, 2004.