University of Alberta

Descriptive Simulation Modeling with Resource Allocation Using Belief Networks

by

Anthony Allan Van Tol ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Masters of Science

in

Construction Engineering and Management

Department of Civil and Environmental Engineering

Edmonton, Alberta

Fall 2000

0-612-59893-4

University of Alberta

Library Release Form

Name of Author: Anthony Allan Van Tol

Title of Thesis: Descriptive Simulation Modeling With Resource Allocation Using Belief Networks

Degree: Masters of Science

Year This Degree Granted: 2000

Anthony Van Tol
776 Saddleback Road
Edmonton, Alberta, Canada
T6J 5E6

Dated: July 26, 2000

Abstract

Project planning is essential to the successful completion of a project. It defines scope of a project and its method of construction. Traditionally, only the project drawings and specifications are transferred from the design engineer to field construction personnel. There is little in the means of a scientific method that defines or communicates the intended construction method and project plan between the parties involved, resulting in an information gap.

This thesis presents a descriptive simulation modeling tool that bridges this information gap. The modeling tool utilizes a modeling structure that facilitates the identification of relationships between modeling elements and thereby allows the determination of construction processes in the model.

In addition, a resource allocation algorithm for the simulation template is presented. It uses a belief network to identify problems in resource allocations. These problems are linked to remedial measures that modify the model to achieve an improved resource allocation.

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled "Descriptive Simulation Modeling with Resource Allocation Using Belief Networks" submitted by Anthony Allan Van Tol in partial fulfillment of the requirements for the degree of Masters of Science in Construction Engineering and Management.

Dr S. M. AbouRizk

Dr. M. Zuo

Dr A. Robinson

Approved on June 21, 2000

Dedication

I would like to thank my supervising professor Dr. Simaan AbouRizk for his time, encouragement, support, and technical guidance during the course of my research. Appreciation is also extended to the members of my supervisory committee, including Dr. Robinson, and Dr. Zuo.

Table Of Contents

List Of Figures

Chapter 2 Descriptive Modeling Approach

Chapter 3 Simulation Modeling Template

Chapter 4 Resource Allocation Improvement Approach

Chapter 5 Belief Networks

Chapter 6 Integration of Simulation and Belief Networks

List Of Tables

Chapter 2 Descriptive Modeling Approach

Chapter 3 Simulation Modeling Template

Chapter 5 Belief Networks

Chapter 6 Integration of Simulation and Belief Networks

Chapter 1

Thesis Scope and Objectives

1.1 Statement of Problem

There are two unequivocal facts in construction management, the first of which is that all construction projects are considered unique from one another. Many varying conditions, such as environment, facility design, project materials, available resources, and available labor skills, all determine the individual characteristics of a project. The second fact is that a construction project is a one-time only undertaking. A facility can only be built once, and a constructor is only given one chance at building that facility as efficiently as possible. Thus, the successful completion of a project relies heavily on its project management.

Project management is defined as the art and science of directing and coordinating human and material resources to achieve stated objectives within the constraints of scope, time, cost, quality, and to the satisfaction of all parties involved (Ahuja, Dozzi, and AbouRizk, 1994). Project planning is one such component of project management vital to the successful completion of a project. Project planning begins with the conception of a project, and is carried on through to project completion. Project planning starts to gain considerable attention at the design stage of a project when the design engineers take constructability concerns into account in their design. Generally, design engineers communicate their design of facilities to the contractors through the method of drawings and specifications. Contractors then build these facilities by interpreting the design documents, determining an approach to its construction, coming up with a conceptual plan for construction, and then leaving the rest to the field personnel. The specific details of the conceptual plan for construction, such as exactly how an activity is to be carried out, is left to the field personnel to determine. Generally speaking, the construction industry can be described as a "trade" or "craft" culture. This implies that, for the most part, the specific details for the construction methods used in a project are left to the field personnel to determine. This process usually occurs in an unstructured manner where there is little in the way of a scientific process.

Thus, the problem at hand is the transfer of information gap that exists between the design created by the design engineer, the conceptual plan for construction determined by the construction estimator, and the construction methods carried out by the construction superintendent and/or Foreman. There is a breakdown in the planning process that occurs as the project is passed between each of these three parties. This causes actions in the original plan to be duplicated in effort, modified, or even forgotten. There is little in the means of a scientific method for communicating the project plan between any of these parties, a method that would minimize errors and inefficiencies. Traditionally, the only pieces of information that are passed on between the three parties are the project drawings and specifications. A solution to this problem is the creation of a tool that can come up with the equivalent of the drawings and specifications, which also facilitates communication with all parties involved, as far as the construction methods are concerned.

It is also generally believed that the ability to influence a project's cost and duration is greatest during the early stages of a project. This belief comes from the evidential fact that, as a project progresses, the cost and duration of that project become dependent on decisions already made. In other words the decisions made at the beginning of a project set the limitations on alternatives for decisions to be made later on in the project. Therefore, it could be stated that with increased precision and efficiency in a project's plan and communication from the design phase forward, better project performance would occur.

Some of the more traditionally used project planning methods have been the Critical Path Method (CPM), the Program Evaluation and Review Technique (PERT), Bar Charts, and Linear Scheduling Methods. CPM details the early start, late start, early completion, and late completion times of activities within a project using a network approach. PERT analyzes the duration of activities within a project by assessing the duration of each individual activity within the project as a random distribution. Bar charts are a graphical plot of the project activities, in terms of their duration, against time. Linear scheduling is used for projects that are linear or repetitive in nature. For this method, sloping lines are created as either a location or a number of units is plotted on the y -axis against time on the x-axis. The slope of the line indicates the expected productivity. Even though these methods have proven their merit, they do have limitations. CPM, PERT, and Bar Charts have are unable to efficiently model repetitive or linear construction processes. All these methods lack the ability to effectively model resource interactions and detail the construction methods to be used. These methods mainly describe the activity durations, productivity, and expected starts and finishes. They lack the ability to communicate the details of the project plan between the designer, estimator, and construction field personnel.

1.2 Research Objectives

The research objectives for this dissertation are two fold. Both problems presented here are intended to attack and improve the deficiencies in the planning process noted in the prior section. The first of these objectives is to create a general purpose simulation modeling template that bridges the communication gap between the design documentation and the construction methodology. The second objective is to further the planning of a project by creating an algorithm to improve the resource allocation used in the construction process described by the modeling template.

As a solution to the first problem, it is the intention of the author to present a prototype general purpose simulation modeling template similar to that created by Halpin for his General Purpose Simulation Tool, CYCLONE. Although this prototype was designed to closely follow that of CYCLONE, it does deal with some of its limitations and therefore has some slight modifications. As a solution to the second problem, the author has delved into the use of belief networks as a decision-making tool to aid in the improvement of the allocation of resources to a project.

The end result of this research is a general Purpose simulation modeling tool that will be able to describe the construction operation being modeled. This model can be created, observed, and modified at any given time during the life of the project. The observer will be able to determine in a step like manner every detail of the construction process being modeled. The modeling tool will also improve the resources allocated to the activities. Thus, this tool will not only spell out the construction method to the user, but also determine the preferred number of resources to be allocated to the project. This tool therefore allows the planning process to occur in greater detail, beginning with the conceptual phase of a project.

The advantage of using simulation as a planning tool is that it allows the construction project, a one time undertaking, to be simulated and conceptually carried out as many times as required until the desired end product is achieved. Simulation allows a project to be carried out with minimal costs and relatively little time in comparison to the actual construction, without any consumption or destruction of resources, and with no compromise to the safety of anyone involved. AbouRizk and Shi (1994), and Paulson (1987) described construction operations as having inherent randomness with dynamic interactions between resources and the processes that utilize them. This is another advantage that simulation provides over the previously mentioned traditional planning techniques. CPM, Bar Charts, and Linear Scheduling make no considerations for the inherent randomness and dynamic interactions between their resources and the processes that interact with them. In these methods, durations are fixed as are the interactions between the activities. While PERT does assess the duration of an activity as a random distribution, it does not capture the inherent randomness and dynamic interactions between resources and the processes that utilize them. Again, as with the other traditional methods, in PERT the interactions between the activities are fixed.

1.3 Research Scope

This prototype model has been built to demonstrate both the descriptive modeling approach used in bridging the communication gap between designer, estimator, and field construction personnel, and also to demonstrate the resource allocation improvement process. The software systems used in this prototype are as follows: the programming language Microsoft® Visual Basic™ 6.0 was used to create a user interface that facilitated the integration of the software; the simulation shell Simphony© Version 1.01, created at the University of Alberta, was used to create the simulation modeling template; Microsoft® Bayes Networks (MSBN™) Version 1.001 was utilized for the creation and evaluation of the belief networks; and Microsoft® Access 2000 ™ was utilized for data storage. The software for this prototype was chosen for two reasons. The first reason was the software's user friendliness, as it is important to have an end product that is quick and easy to use. If the interface presents itself as being too complicated and time consuming, its effectiveness is diminished. Thus, the user requirements for operating this prototype are kept simple and object-orientated. The second reason for the choice of this software is its flexibility in regards to its integration using Microsoft® Visual Basic™ 6.0.

This research focuses on the creation of a descriptive modeling template that will use a resource allocation improvement process to determine the resources required for the project. For this modeling template, the interaction between the entities within the model has been seen as a server-customer relationship, as used in queuing theory. For example, in a situation where a crane is being used to unload beams from trucks, if the interaction between the crane and trucks is being modeled, the crane would be viewed as a server and the trucks would be viewed as customers. Similarly, in a steel fabrication process, the pieces of steel would be customers and the welder would be a server.

Although the approach developed here is generic, the current prototype has been created for cyclic operations. This does not imply that this method only applies to these operations. Other simulation models that are linear in nature and do not express a cyclic nature, such as a steel fabrication process or a gravel crushing operation, can be improved in a similar manner. This prototype could be improved upon to allow for such models, but for the moment the scope of this research will focus on cyclic operations. The modifications to this prototype that would allow such modeling will be discussed later on in the conclusion of this thesis.

1.4 State of the Art

This research deals with a descriptive general purpose modeling template and a resource allocation improvement method for use as a planning tool. For clarity, the state of the art for the topics involved will be introduced before the topic is discussed. In chapter 2, section 2.2 will review the state of the art for descriptive simulation modeling. In chapter 4, section 4.2 will review the state of the art for simulation resource allocation improvement techniques. In chapter 5, section 5.2 will review the state of the art for belief networks.

1.5 Overview of Approach

This prototype consists of 4 software programs. Simphony© Version 1.01 is used to create and evaluate the simulation model. It is within Simphony that the user will create the simulation model and define its parameters. Using the modeling elements within the modeling template, the user has the flexibility to model any situation desired. The parameters, which are defined in each of the elements, define the characteristics of the model and how it performs. Microsoft® Bayes Networks (MSBN™) Version 1.001 was used to create and evaluate the belief networks used for the resource allocation improvement. There are a few constraints that the user must follow when creating a belief network, but this prototype has been designed to allow the user some flexibility as to the desired construction of the belief network. The Microsoft® Visual Basic™ 6.0

interface has been created such that it is quite simple and leaves little room for user error. The interface requires the user to pick the simulation model file, pick the belief network file, provide information as to how the recommendations from the belief network are to be interpreted for manipulation of the simulation model, and finally how the user wants to control the resource allocation improvement process. The user has the ability to observe and control every iteration, or to let the computer automate the process.

The solution to the first problem, the creation of a tool that is the equivalent of the construction drawings and specifications while facilitating communication with the field personnel, is to be solved by creating a modeling template that will describe itself. That is, a tool able to create a simulation model that will be able to communicate what activities are occurring, what order these activities are occurring in, and what resources are required to carry these activities out.

A solution to the second problem stated is to create an resource allocation improvement method that utilizes performance indices, that describe the state of the simulation model for a given state of resources, and to use a Belief Network as a decision-making tool to decide how to modify the current state of resources. Thus the prototype that is presented here will run the simulation of a model, get the performance indices from the model, use these indices as input to the belief network, and then alter the simulation model based on the results from the belief network.

1.6 Summary

In this chapter, the research has been introduced, two problems have been stated, and an introduction to their solutions has been given. The first of these problems is to create a simulation modeling template that can come up with the equivalent of the drawings and specifications used in the construction of a facility, while providing a means of communicating the project plan to all parties involved. The intent here is to create a structured scientific method through which the designer, estimator, and constructor can communicate their planning process in terms of the facility construction. The second problem stated was to arrive at a generic means of improving resource allocation for a modeled project.

Background on these two problems was given as was the approach to their solution. For the first problem, a modeling template was to be created that would have the ability to describe itself. That is, the model created would have the ability to describe the activities within itself and the characteristics of those activities. The solution for the second problem was to create a resource allocation improvement algorithm that made use of belief networks as a decision-making tool.

1.7 Thesis Organization

This dissertation is organized into the following chapters: chapter 2 explains the descriptive modeling process; chapter 3 explains the simulation modeling template and the modeling elements within it; chapter 4 describes the algorithm and general approach to the resource allocation improvement process; chapter 5 introduces the reader to Belief Networks, how they work, and how they are to be used in this research; chapter 6 deals with the integration of the resource allocation improvement algorithm and the belief network and also contains all the information regarding the validation of the modeling tool; and chapter 7 is the final summary chapter of this dissertation and contains the conclusion, contributions, and recommendations for this research.

Chapter 2

Descriptive Modeling Approach

2.1 Introduction

There is a noticeable gap or breakdown in the transfer of project planning information as a project travels from the hands of the designer to the estimator to the field construction personnel. This breakdown can lead to duplication of effort in the planning of the project, and possible errors and changes to the original plan. Duplication of effort most often occurs between the estimator and the field construction personnel. The estimator goes through the project initially and estimates the project as he or she perceives it being carried out. They have now defined the manner in which the construction process is to occur. They have determined the required resources for the project and how they should be allocated to the activities within the project. In most cases, once the project is turned over to the field construction personnel, the project engineer and superintendent redo this process of determining the construction methods and resource allocations on a weekly or monthly basis. Using an earthmoving example, an estimator may have assigned five excavators to a project and has assigned each of these excavators to perform specific activities within the project so that it is completed in the most cost-effective means possible. The construction superintendent, not knowing the intentions of the estimator, may reassign these excavators to activities that the estimator had not intended them to do. Now not only has the superintendent duplicated the estimators' efforts by reallocating the excavators to the activities in the project, but he may have done it differently than originally planned. This might result in increased costs, which is considered an error, and limited the options for decisions to be made in the in the future by making a change to the original plan.

Increased effort and detail in project planning can only improve construction performance. Therefore, a useful means of communicating the project plan between everyone involved in the construction process is a descriptive modeling tool. Project drawings and specifications only detail what is to be built and the tolerances or allowable deviations the built facility has from the designed facility. The advantage of a descriptive modeling simulation tool is the manner in which the tool spells out how to construct the facility. If the construction process or system is modeled correctly, the described method should meet the specifications outlined by the designer.

Once again, the advantage of a simulation is its ability to conceptually carry out a project time and time again for a variety of scenarios until the most effective plan of attack for building the facility is found. Once the baseline model has been created, modifications to the model can occur at any time during the life of the project. Change orders, additional work, and changes in the construction method can be introduced as they occur in the project. The modeler can introduce limitations and constraints to the model as they are experienced in reality. The user can also, at any time during the project, evaluate, through simulation, alternatives and options to construction before making a final decision.

Lastly, when creating the modeling template, it was the author's intention to create a tool that was quick and easy to use. The modeler therefore needs little knowledge in any programming languages, as, for the most part, the construction of the model is visual: the modeler needs only to physically place and link elements until a model is created. The only non-visual part of this process is the definition of the attributes for the elements. Ideally, the user needs no knowledge in programming, but Simphony allows the user to define parameters using Visual Basic code. The author sees this feature as a huge advantage of the Simphony simulation shell, as it allows the user to push the modeling tool to new limits by letting parameters be dependent on other modeling element parameters and states of the model during simulation. For this feature to be effectively used, however, the user would require a good understanding of the modeling template in order to know the variables and how to manipulate them. For the most part, this feature is not required for descriptive modeling. Overall, Simphony should be readily useable by anyone not disciplined in computer simulation.

This chapter is broken down in the following manner: section 2.2 gives insight into the previous contributions put forth in the area of simulation model description; section 2.3 gives an overview of the approach taken to create a method of describing a simulation model; section 2.4 presents the algorithm used to carry out the description of a simulation model; section 2.5 explains the organization required for a descriptive simulation model; section 2.6 explains how the activities are identified and used in the descriptive modeling process; section 2.7 explains how the resources are identified and used in the descriptive modeling process; section 2.8 explains the outputs from the descriptive modeling process; and finally section 2.9 summarizes the chapter.

2.2 State of the Art

Duokidis and Paul (1985) developed a system that addressed the need for an improved method of simulation model development. Their intention was to decrease the time involved in the formulation phase of a simulation model. They tried two approaches to solve the problem at hand, the first of which was a Production System approach that made use of an expert system. This system used factual knowledge and heuristic rules to draw information from the user to define the model. The model was defined by identifying the activities within it, the relations between the activities, the model entities, and the paths for the entities to follow. The downfall of this system was its rigid nature. First, the system had a terminology applicable only to the simulation analyst. Second, each activity was subject to the same if/then questions that defined its respective role(s) in the simulation. This resulted in little or no improvement in the formulation time required to create the simulation model. Lastly, constraints imposed on the scope of the system did not readily allow the expert system to be applied to the solution to the problem. This rigidity resulted in the second approach of using a Natural Language Understanding System. This approach allowed the user to define the actions in the model by using english sentences as inputs. The sentences used as inputs were simple sentences that described the actions occurring in the model, such as "A loader loads trucks". The user would thus create a dialogue, which was controlled by the computer, to define the model at hand. This system was found more advantageous than the previous one, since it was more interactive with the user and it operated in a broadly applicable language.

One of the earlier and better known Discrete Simulation systems used for modeling construction projects is CYCLONE (Vanegas, Bravo, and Halpin, 1993). This simulation tool made use of seven basic modeling elements, which effectively created a system able to model cyclic operations, such as tunneling, earthmoving, and mix plant operations. This tool built a foundation for other simulation systems, such as CIPROS, and serves as the basis for this dissertation's modeling template.

Tommelein, Odeh and Carr (1994) developed CIPROS, an object orientated and interactive system that integrated project- and process-level planning. It was a simulation tool that modeled construction processes by matching resource properties with design component properties and operation durations. In CIPROS, the user would identify the attributes of the facility to be constructed from the project drawings and specifications, develop a CPM for the project, and then pick a construction method for each activity from a library of pre-developed simulation activity model networks. CIPROS then pieced together these networks based on the sequential relationships the activities share, defined by the information from the CPM, design drawings, and specifications. The user would then specify the resources available to the project and which activities could share which resources.

Davidson (1988) used an expert system as a diagnostic tool for a polymerization process to make recommendations for corrective actions, and explanations for causes relating to off spec polymer production. Although this expert system did not describe or create the polymerization process, it was able to determine, through a procedure of asking questions and receiving answers from the user, the current state of the process and make suggestions for modifications to the process in order to improve production. Helweg and El - Khashab used an expert system in a similar manner to aid in canal design. The expert system they created took into account canal function, hydraulics, operation, construction, and economics. In terms of construction, this expert system minimized the lining costs, canal lengths, crossing structures, and maintenance requirements. Earthworks were viewed as a major cost of the canals, and considerations were therefore made in terms of excavation, fill, compaction, hauling, trimming, and material export and import. This expert system did not define or detail the construction process for the canal but took into account the construction process for design.

Papamicheal and Selkowitz (1990) created a Building Design Support Environment (BDSE) that supported building design from its conceptual phase to its construction, occupancy, and use. The system consisted of imaging, simulation, and expert system software linked in a multimedia environment containing the resource information of handbooks, product catalogs, and case studies. For the most part, the literature describing this system focused on building design, and gave no reference as to how it facilitated the determination of a construction method for the building.

Tommelein et al. (1991) developed SightPlan, which was a knowledge-based expert system used to plan the layout of temporary construction facilities. This expert system mimicked the manner in which a construction manager would determine the layout of the temporary construction facilities for a project. The output from this system was the suggested manner in which the site was to be laid out for construction.

Reinschmidt, Griffs, and Bronner (1991) created a Construction Management Display System (COMANDS). This system was comprised of a 3-D modeling tool, a relational database used to store information for objects in the models created within the 3-D environment, and the scheduling software program Primavera. This tool has proven beneficial in letting construction personnel investigate procedures for construction. Primavera and the relational database allow individual objects within the model to be assigned work packages such that time dependant snap shots of the model and its progress can be taken. Using this tool, design errors and constructablitiy concerns are easily identified, and improved scheduling capabilities are provided for.

Tiecholz and Fischer (1994) identified how design and construction of capital projects are characterized by fragmentation and paper-based exchange of information. Their solution to this problem was a shared object-orientated project model that integrated all aspects of the construction process from design to construction to use and maintenance. This system consisted of a core project model, created from 3-D CAD objects, that was integrated to all other software programs used in the project. Thus, the 3-D CAD model facilitated the communication between all parties involved in the project. The 3-D model at the core of this system was created using predefined symbolic objects that were linked to a database in which information about the objects was stored.

Kartam (1994) also identified fragmentation between the architecture, engineering, and construction industry. His approach to the solution was also through the use of 3-D modeling. His creation, Intelligent Computer-Aided Design System (ISICAD), was a CAD interface that made use of object-orientated modeling to create 3-D models that would be integratable with databases used to incorporate project data and expert systems. This system was found beneficial when used as a means of communicating and evaluating design, as well as resolving conflicts that arose due to design decisions. Fragmentation between the parties involved was therefore reduced, facilitating better communication of the design. In the prototype system, ISICAD consisted of the CAD interface, a construction scheduling knowledge based system, and a constructability improvement knowledge based system.

Navon (1995) identified the inefficiency in the manner in which designs are currently communicated between the designer and contractor. He stated that the process focuses on the submittal of documents between the two parties: the designer sends the contractor drawings and specifications and the contractor sends the designer submittals to demonstrate his understanding of the drawings and specifications. Navon stated that this process contributes to fragmentation within the industry, impedes communication, obstructs understanding, and leads to change orders and litigation. He felt that Automatic Electronic Data Flow between the design, construction planning, and onsite construction phases of a project was the key element in computer aided construction, and a solution to this problem.

Vaugn (1996) saw inefficiencies in the way the design was communicated from the designer to the contractor. Therefore, as a solution to this problem, he built a 4-D modeling system that incorporated the 3-D CAD modeling system with a 4th dimension of time, a tool that he saw advantageous to the design-build process. The 3-D CAD modeling articulated potential problems better than disjointed 2-D drawings, and the 4th dimension of time was beneficial in avoiding construction constraints and scheduling conflicts.

Lastly, Trego (1997), in her article "Computers in Manufacturing", outlines the current state of Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) software, and how simplifying and automating the design process while enabling better communication of designs and specifications between the designer and manufacturer has proven advantageous. 3-D modeling tools, designed specifically for the machining process, can accurately generate process plans for the machined components. In her article she looks at the commercial software programs available and comments on each of their abilities. These programs are CATIACAM, Pro/ENGINEER, Euclid Quantum, SURFCAM, and Varimetrix.

2.3 Overview of Descriptive Process

The descriptive simulation modeling method using this template identifies the activities and resources used in a project and how they are related to each other. The actual modeling process used to create a model is similar to that of most other computer simulation tools. Construction of a model mainly involves placing modeling elements, which are represented by small objects or icons, and relating the elements to each other by linking them together (Figure 2.1). The definition of the element's properties is done by assigning attributes for its parameters. (Figure 2.2)

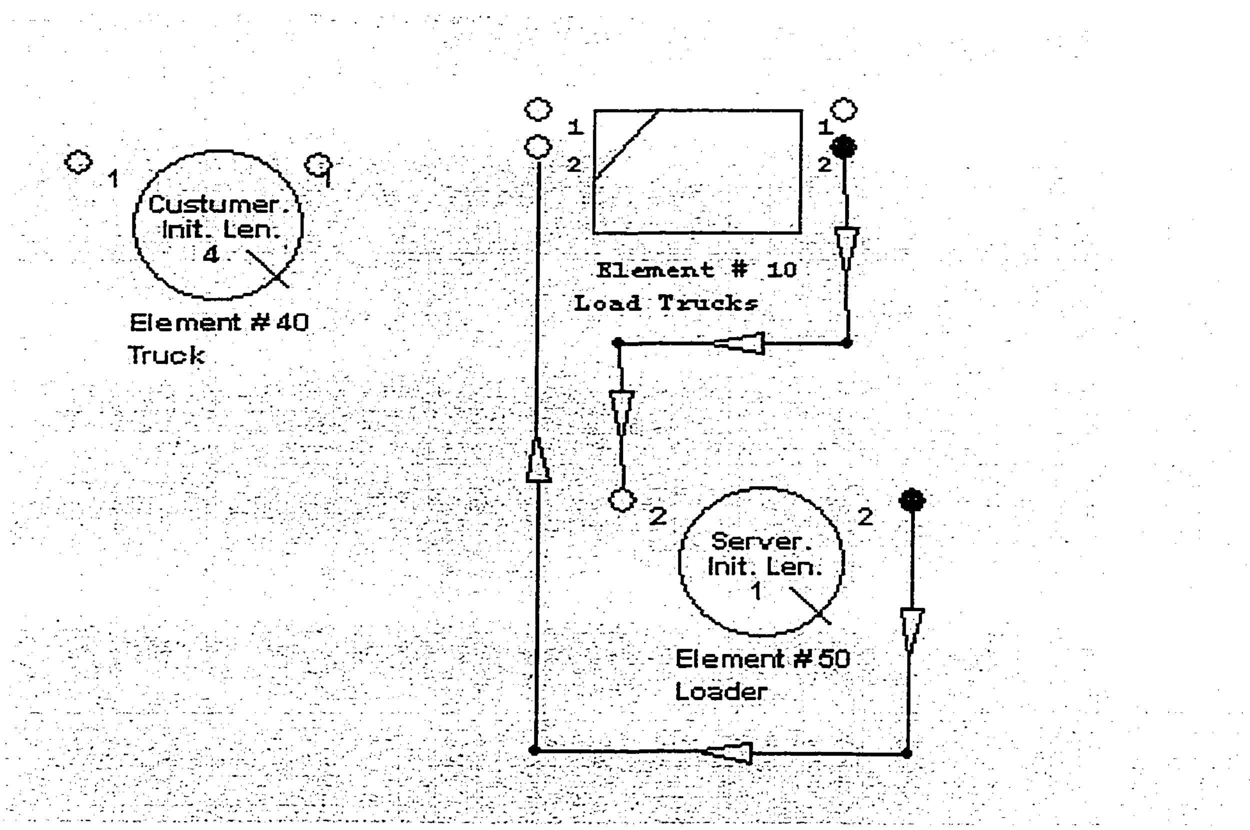


Figure 2.1 Simphony Model In Progress

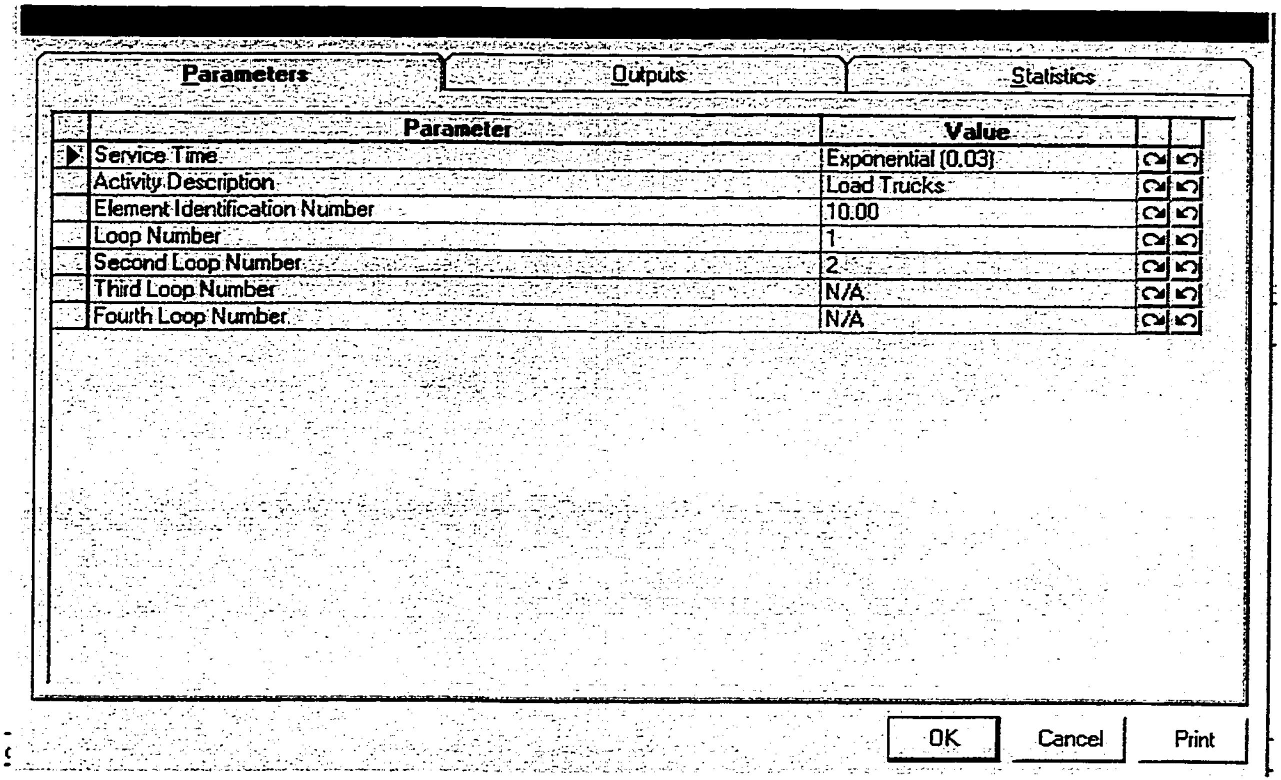


Figure 2.2 Combi Modeling Element Parameters

The template created for this thesis is a general purpose simulation modeling tool. The modeling elements used in this template are generic, and it is the definition of their attributes that gives them their characteristics. An element representing a resource is just an element representing a resource until the user defines that element to be a loader, a labor crew, or crane. Special purpose simulation modeling tools are domain specific to the extent that the modeling elements used for modeling actually represent the activities or resources. For example, in an earthmoving special purpose modeling template, there might be a modeling element that is called a truck that would represent the truck resource in the model. Another element might be a dump location that would represent the dumping activity for the truck. One can see that the special purpose simulation tool is simpler and easier to use, but that comes as a trade off for its limited flexibility and ability to only model within its domain. This template, being a general purpose modeling template, provides the flexibility to model in more than one domain.

This general purpose simulation modeling template closely resembles and follows the modeling methods that are used in CYCLONE. While developing this prototype, some of the limitations of CYCLONE have been experienced. Therefore, some adaptations were made to the template that make it slightly different from that of CYCLONE. One such adaptation is the creation of a modeling structure that the user must follow when modeling. This structure deals with the identification of cyclic loops within the model, and the identification of the order of the activities within these loops. Other adaptations include the creation of additional parameters for the modeling elements. Activity descriptions are now required for activity modeling elements. Resource elements require resource descriptions. The resource allocation improvement routine, which is the second objective of this research, requires directed entities in the model and additional parameters used for the evaluation of the simulation model and operation of the resource allocation improvement algorithm. The modeling template and its elements will be introduced in detail in the next chapter. It should be noted that CYCLONE was used only to demonstrate the theory presented in this thesis. This research should not be viewed as a contribution to the CYCLONE modeling tool.

It should be noted that the elements used in this prototype template have been created with a maximum of four connection points. This is to allow the user to connect each element to four different loops within the simulation model but also places a limitation on the capabilities of the modeling template. CYCLONE allows for an infinite number of connections from its elements, making it more flexible for modeling in this regard. Due simply to the complexity and time required to develop a template that would perform this feat, the limitation of only four connection points has been imposed. Even though the elements are limited in their connection points, the concept being presented here is clearly defined. The capability of infinite connection points will be left for future efforts in this area of study.

2.4 Algorithm

The very first step in the descriptive modeling process is the creation of a simulation model following the loop and element identification structure. The first step to this process is to identify the cyclic loops within a model; these loops are the cyclic paths that entities follow. Unlike CYCLONE, where the entities were anonymous and free to follow whatever path they wanted, this modeling template has directed entities. In this modeling template, the entities are directed and restricted from traveling outside of their loops.

In an earthmoving modeling example, there could be three loops identified (Figure 2.3): the loader loop, the truck loop, and the spotter loop. The loader loop would consist of the cyclic activities performed by a loader, such as "Load a truck", "Clean up immediate area", and "Prepare for the arrival of the next truck". The truck loop would consist of the cyclic activities performed by a truck, such as "Load a truck", "Haul", "Dump", and "Return". The spotter loop would consist of the cyclic activities performed by a spotter, such as "Spot a dump", "Dump", "Spread dumped load", "Clean up immediate area", and "Get positioned for the next truck". Now, since the entities are directed, an entity created in the truck loop will remain in that loop for the duration of that simulation. It is unable to move into another loop; thus, a truck entity cannot travel in the loader loop. The modeling elements can have up to four connection points, and they can therefore be part of four different loops. For example, the "Load a truck" activity in the previous example is part of both the truck loop and loader loop.

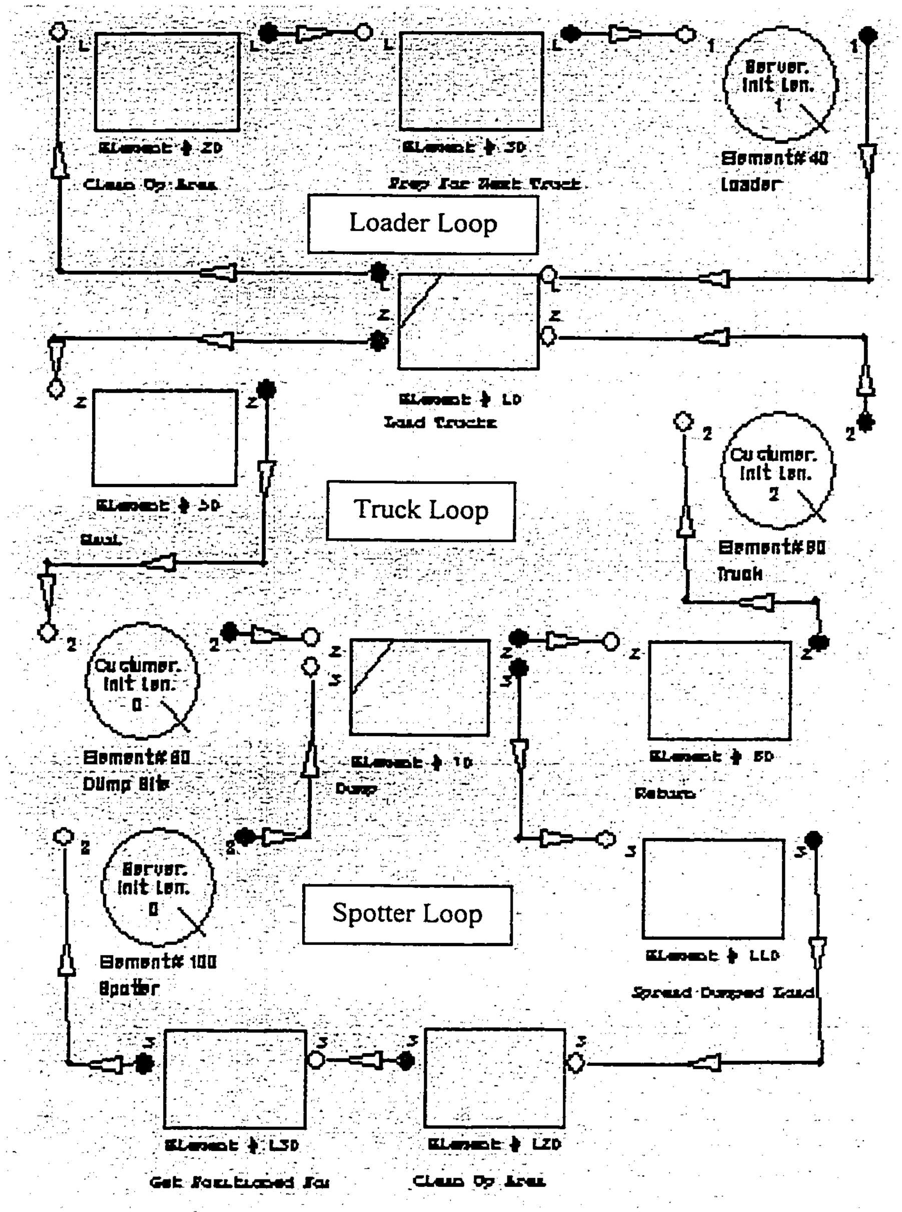


Figure 2.3 Earthmoving Model

The modeling element identification number assigned to each modeling element is used to assign priority or position to the modeling element within its loop. As each element is created, it is assigned an identification number. The first element created receives an identification number of 10, and each successive element created after that will receive an identification number incremented by ten. The identification number serves two purposes, the first of which is to assign an element's priority over other elements. In the case where two activities are competing for a shared resource, the element with the lower identification number will have priority over the other element(s). This allows the user to create simulation models in which resources are shared among several activities where certain activities take priority over other activities. For example, in Figure 2.4, Combi Element #20 will have priority over Combi Element #30 and therefore receive entities from Queue Element #10 first. This is the same as in CYCLONE. The second purpose of the identification number is to assign a position to the modeling element in its loop. The elements are to be placed in the loop with identification numbers ranging from lowest to highest, starting with the very first activity to occur in the simulation.

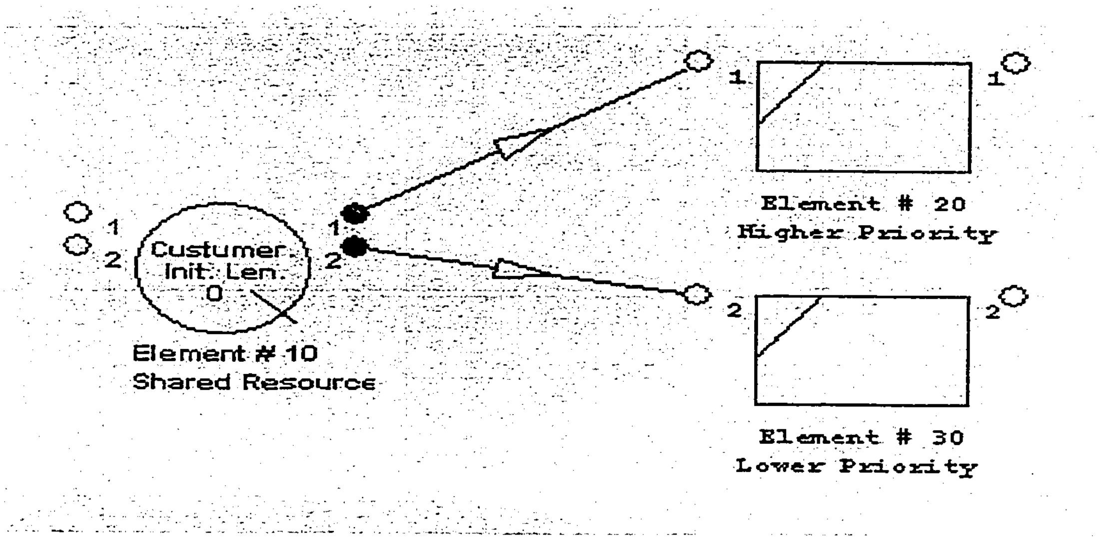


Figure 2.4 Priority For Combi Modeling Elements

As an example, a simple truck cycle in an earthmoving operation has been described (Figure 2.5). Basically, this cycle consists of a truck being loaded, the truck's haul, the dump, and the truck's return back to the loader. To model this, the user would create a model where the truck being loaded element has an identification number of 10. The haul activity would have an identification number of 20. The dump would have an identification number of 30. The return would have an identification number of 40. The Queue Element with an identification number of 5 is added to the loop so as to add the trucks to the model.

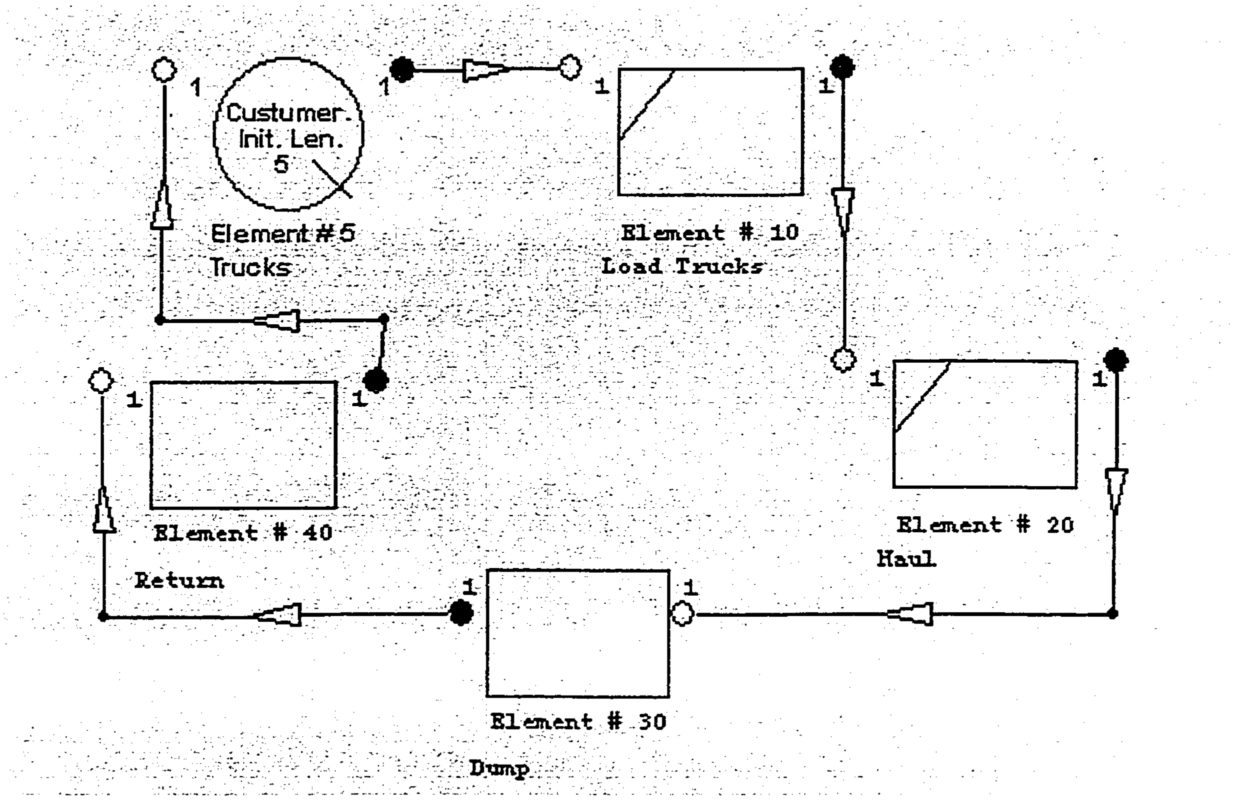


Figure 2.5 A Modeled Truck Cycle

Once a model has been built, the algorithm for describing the model is used (Figure 2.6). The first step in the process is the reading of the model and the recording of the loop number(s) and identification number for every element in the model (Table 2.1). Since each element has up to four connection points, four loop numbers are recorded. A value of "N/A" is used where no connection point is attached to a loop. These values are recorded in a hidden table within Simphony.

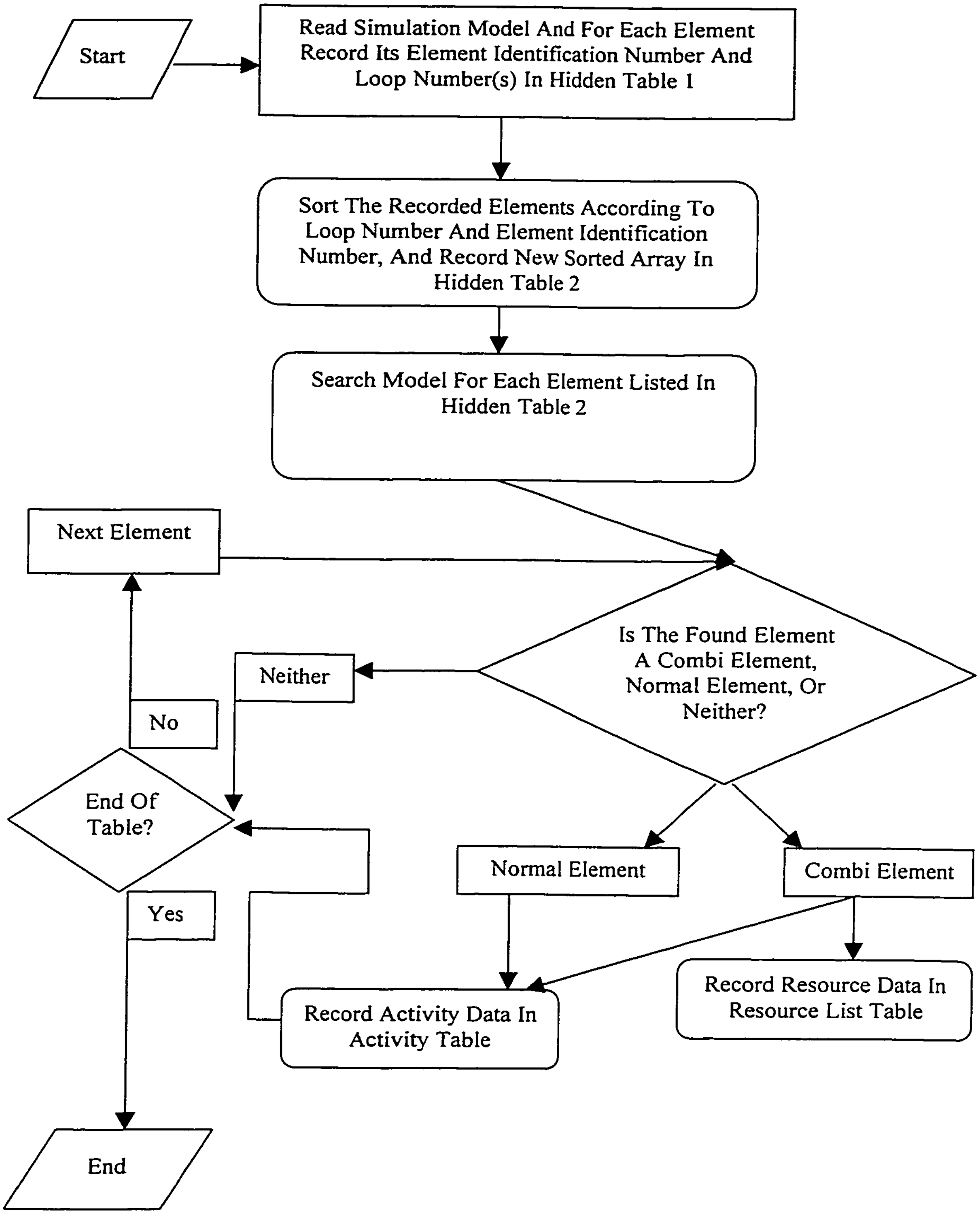


Figure 2.6 Descriptive Modeling Algorithm

	10		1	2	N/A	N/A
2	20		1	N/A	N/A	N/A
3	30		1	N/A	N/A	N/A
4	40		1	N/A	N/A	N/A
5	50		2	N/A	N/A	N/A
6	60		2	N/A	N/A	N/A
7	70		2	3	N/A	N/A
8	80		2	N/A	N/A	N/A
9	90		2	N/A	N/A	N/A
10	100		3	N/A	N/A	N/A
11	110		3	N/A	N/A	N/A
12	120		3	N/A	N/A	N/A
13	130		3	N/A	N/A	N/A

Table 2.1 The Initial Element Information

From this initial hidden table another hidden table is created in Simphony (Table 2.2). In this table, for every loop in the model, the elements within it are listed. Therefore if an element is part of two, three, or four loops it will appear in the new table two, three, or four times respectively. After all the entries have been made into the new table, the elements are sorted within their loops. That is, for loop 1, all the elements within that loop are sorted from lowest to highest. Assuming the user followed the constraints of element priority with the element identification numbers, the list of elements should be in the order in which they occur for the construction process.

	10	1	1	2	N/A	N/A
2	20	1	1	N/A	N/A	N/A
3	30	1	1	N/A	N/A	N/A
4	40	1	1	N/A	N/A	N/A
5	10	2	1	2	N/A	N/A
6	50	2	2	N/A	N/A	N/A
7	60	2	2	N/A	N/A	N/A
8	70	2	2	3	N/A	N/A
9	80	2	2	N/A	N/A	N/A
10	90	2	2	N/A	N/A	N/A
11	70	3	2	3	N/A	N/A
12	100	3	3	N/A	N/A	N/A
13	110	3	3	N/A	N/A	N/A
14	120	3	3	N/A	N/A	N/A
15	130	3	3	N/A	N/A	N/A

Table 2.2 The Sorted Element Information

Once that is done, the model is then searched again for the elements listed in the sorted table. The search starts by looking for the element with the lowest identification number in the first loop. It is followed by the next lowest element and so on until the entire list in the sorted table has been covered. Remember, some elements may have been part of two loops and will be searched for twice.

Once an element has been found, it is checked to see if it is either a Combi Element or a Normal Element; these are the rectangular simulation modeling elements in which activities take place (Figure 2.7). These elements contain the information needed to describe the simulation model. The characteristics of these modeling elements and their differences will be described in the next chapter.

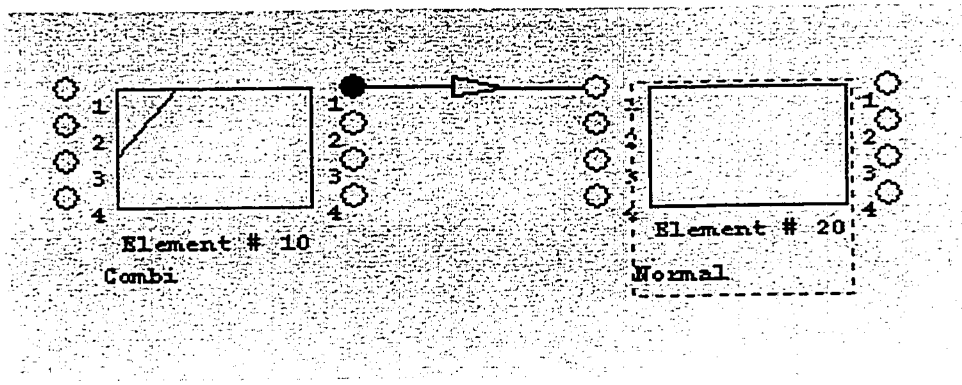


Figure 2.7 The Combi And Normal Modeling Elements

If the found element is a Normal Element, its loop number, activity description, and element identification number are recorded in the Activity Table (Figure 2.8). The loop number that is recorded for the entry into the Activity Table is the loop number for the current entry in the sorted table. Remember that an element can be recorded in this table up to four times with different loop numbers.

	Activity	Loop Number	Resource 1	Resource 2	Resource 3	Resource 4	Activity Node Number
1	Load Trucks	1	Loader	Truck			10
2	Clean Up Area	1					20
3	Prep For Next Truck	1					30
4	Load Trucks	2	Loader	Truck			10
5	Haul	2					50
6	Dump	2	Dump Site	Spotter			70
7	Return	2					80
8	Dump	3	Dump Site	Spotter			70
9	Spread Dumped Load	3					110
10	Clean Up Area	3					120
11	Get Positioned For Next Truck	3					130

Figure 2.8 The Activity Table

If the found element is a Combi Element, in a manner similar to the Normal Element, the Combi element's loop number, activity description, and element identification number are recorded in the Activity Table, as are the resources required to carry out the Combi activity. The resources are found by finding the connected Queue Elements that precede the Combi element. The Queue Elements contain the information for the resources to be used in the Combi activity. Each resource is added to the Resource List Table by adding the queue element identification number, the loop number, the resource description, the activity element identification number, and the activity description (Figure 2.9).

	Loop Number	Queue Node Number	Resource	Activity Node Number	Activity Description	Shared With	Shared With	Shared With
1	2	90	Truck	10	Load Trucks	Queue # 40 - Loader		
2	1	40	Loader	10	Load Trucks	Queue # 90 - Truck		
3	3	100	Spotter	70	Dump	Queue # 60 - Dump Site		
4	2	60	Dump Site	70	Dump	Queue # 100 - Spotter		

Figure 2.9 The Resource List Table

After a resource is entered into the Resource List Table, a search of the Table occurs to see if the same activity occurs twice or more. If it does, it means that the current activity requires more than one resource to occur. Thus, for each resource that requires other resources to complete an activity, the other resources are listed in the "Shared With" column of the Resource List Table.

2.5 Model Organization

As stated before, the organization of a model follows a loop and element ID number system. Each element is assigned an element identification number that the user has the ability to change. The elements are placed with increasing element identification numbers in their loops in the order they occur during the construction process, and the loop defines the path an entity follows as it progresses through its activities.

The following example is one Haplin used to model an earthmoving operation (Figure 2.10). The first step to modeling this operation is to understand the process and identify the number of cyclic loops within the model. This model has 5 cyclic loops: the dozer cycle, the loader cycle, the truck cycle, the spotter cycle, and the spreader dozer cycle. There is also one path that the soil follows as it travels from its source to its placement. This can be called a loop also even though it is not cyclic. The algorithm will still be able to describe the activities that the soil has to go through in the modeling process. Each loop or path contains all the activities for the entity (or entities) that travel along them. The dozer cycle contains the stockpile soil activity. The loader cycle contains the load activity. The truck cycle contains the load, haul, dump, and return activity. The spotter cycle contains the dump activity. The spreader dozer cycle contains the spread dirt activity. The soil's path contains the activities of stockpile soil, load, haul, dump, and spread dirt.

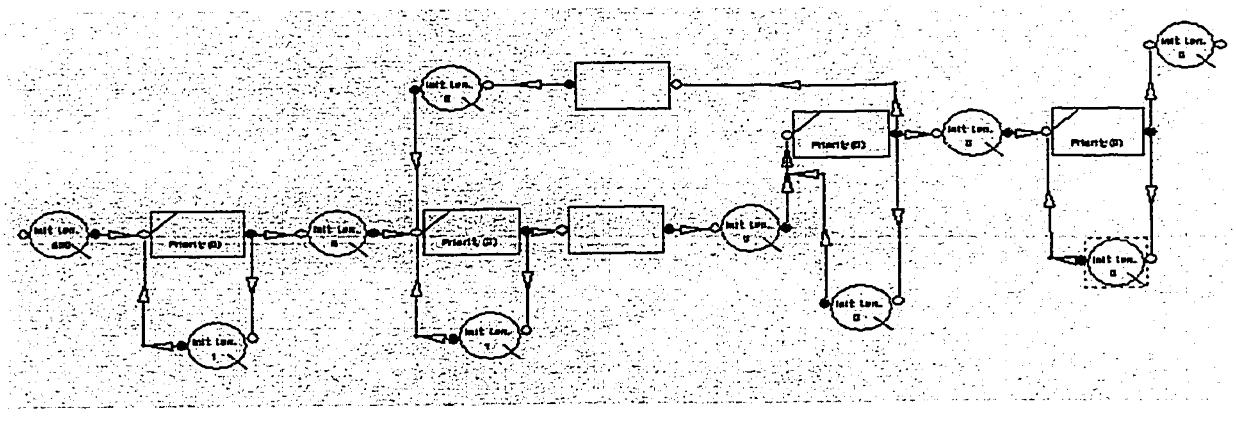


Figure 2.10 CYCLONE Model Of An Earthmoving Process

For every Combi activity, there are resources required for it to occur. In Halpin's model, there are four Combi modeling elements used. For the stockpile soil Combi activity, the resources of the dozer and the soil are required. For the Load Combi activity, the loader, stockpiled soil, and truck are required. The dump Combi activity requires a truck, a load in the truck, and a spotter to occur. The spread dirt Combi activity needs the dumped load and the dozer for its activity completion.

The modeler now has all the information required to model the earthmoving operation. The number of loops have been defined, the activities within the loops have been defined, and the resources required for the Combi activities have been defined. Now all the user must do is begin placing the modeling elements such that their Element Identification Numbers are in order from lowest to highest as they progress through the construction process.

To model the same operation Halpin has described in Figure 2.10, one would do the following. The first step would be to assign the loops numbers from 1 to 6 for the soil's path, the stockpile cycle, the loader cycle, the truck cycle, the spotter cycle, and the spreader dozer cycle respectively (Figure 2.11). The next step would be to begin placing modeling elements starting with the very first modeling element required in the simulation. For this model, that element would be the Queue Element for the ground available. The user would place this element within loop number 1, and its element identification number would be 10. The next element to be placed would be the stockpile soil element, with its loop numbers being 1 and 2. This element is automatically assigned the element identification number of 20. The next element to be placed after that would be the Queue Element for the dozer doing the stockpiling; its loop number would be 2. The next element to be placed after that would be the load Combi Element with its loop numbers being 1, 3, and 4. This process would go on until all the elements were placed and their element identification numbers and loop numbers are as in Table 2.3.

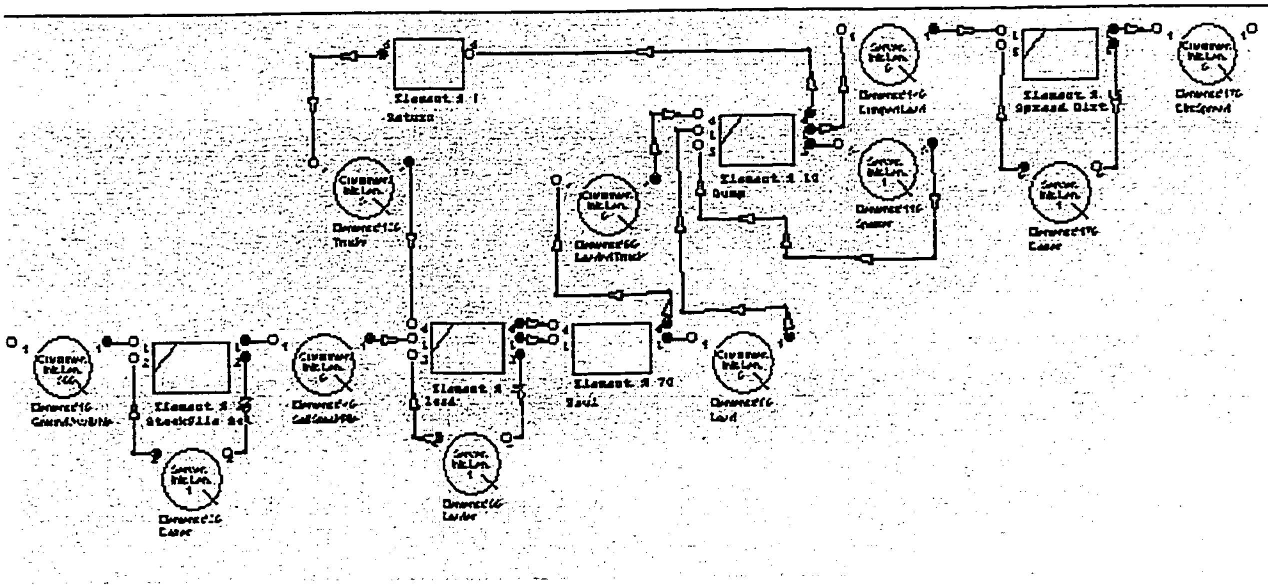


Figure 2.11 Descriptive Cyclone Model Of An Earthmoving Process

Element	ID Number	Loop Number 1	Loop Number 2	Loop Number 3	Loop Number 4
Ground Available	10	1	N/A	N/A	N/A
Stockpile Soil	20	1	2	N/A	N/A
Dozer Idle	30	2	N/A	N/A	N/A
Soil Stockpile	40	1	N/A	N/A	N/A
Load	50	1	3	4	N/A
Loader Idle	60	3	N/A	N/A	N/A
Haul	70	1	4	N/A	N/A
Loaded Truck Queue	80	4	N/A	N/A	N/A
Load Queue	90	1	N/A	N/A	N/A
Dump	100	1	4	5	N/A
Dump Spotter Idle	110	5	N/A	N/A	N/A
Return	120	4	N/A	N/A	N/A
Truck Queue	130	4	N/A	N/A	N/A
Dumped Loads Queue	140	1	N/A	N/A	N/A
Spread Dirt	150	1	6	N/A	N/A
Dump Dozer Idle	160	6	N/A	N/A	N/A
Dirt Spread	170	1	N/A	N/A	N/A

Table 2.3 Earthmoving Model Element Identification Numbers And Loop Numbers

The only difference between this model and Halpin's model is the division of the Loaded truck Queue Element just after the haul activity in Halpin's model (Figure 2.10) into the Loaded Truck Queue Element and the Load Queue Element in the descriptive model (Figure 2.11). The reason for this division is the Simphony template directs the entities within the model and cannot combine and then separate entities. Thus, the truck and Load must travel together as separate entities.

2.6 Identification of Activities

As stated before, activities in a model can occur in two different elements, the Combi Element and the Normal Element. These modeling elements will be discussed in detail in the next chapter. For now, it should be understood that the difference between the two elements is that the Combi Element requires a resource, represented by a queue element, directly before it. The Queue Element is where the resource will wait, if it is not already in use in some other portion of the model, until called upon by the Combi Element. The Normal Element activity just lets entities pass through as they come to it.

A Normal or Combi Element is identified by its modeling element name. Referring back to the algorithm used for the descriptive process (Figure 2.6), when the model is being searched a second time, the program is searching for the listed elements and checking to see what their modeling element name is. In the case of this modeling template, the program is looking for "DC_Combi" to indicate a Combi Element and "DC_Normal" to indicate a Normal Element. Once either of the two modeling elements is found, the program knows what kind of element it is and what information to acquire.

The information drawn out of the Normal and Combi Elements are the loop number, the activity description, and the element identification number. The loop number is used to identify what loop in the model the element belongs to (a Combi and Normal Element can belong to up to four loops). The activity description is a phrase that describes the activity occurring at the element. These descriptions can be phrases such as "Build Forms", "Haul", "Inspect Compaction Density", or "Let Concrete Cure". The element identification number identifies the element's location within the loop.

2.7 Identification of Resources

A Queue Element is required to precede each Combi Element. Thus, when a Combi Element (that is an element named "DC_Combi") is found, the Queue Elements connected to it are seen as resources for that activity. To identify these resources, the program goes from the Combi Element and traces back all the incoming relations, or arrows, to the preceding Queue Elements. Once at the Queue Elements, the resource information is taken. This information includes the Queue Element identification number, the loop number, the resource description, the activity element identification number, and the activity description. The Queue Element identification number and loop number are used to identify the modeling element, the resource description is a phrase such as "Loader" or "Crane" that describes the resource within the queue, the activity element identification number is used to identify the Combi Element, and the activity description is the phrase in the Combi Element used to describe the activity. All this information is recorded in the Resource List Table. After each entry in the Resource List Table, a search is made of the table to see if the activity for the current resource appears with any other resources. If it does, the additional resources required to carry out the activity are listed with the current resource in the "Shared With" columns. Thus, for every resource listed in the table, its associated activity is listed along with the additional resources required to carry out that operation.

2.8 Outputs From Descriptive Process

There are two basic output tables from the descriptive modeling process: the Activity Table and the Resource List Table. The Activity Table describes the activities in the model as they occur in their loops (Table 2.4). The Resource List Table links the project resources to their activities while identifying the additional resources required (Table 2.5). The information in the tables in Table 2.4 and Table 2.5 is taken from the descriptive model of an earthmoving process modeled in Figure 2.11.

	Activity	Loop Number	Resource 1	Resource 2	Resource 3	Resource 4	Activity Node Number
1	StockPile Soil	1	Ground Available	Dozer			20
2	Load	1	Trucks	Soil StockPile	Loader		50
3	Haul	1					70
4	Dump	1	Loaded Truck	Load	Spotter		100
5	Spread Dirt	1	Dumped Load	Dozer			150
6	StockPile Soil	2	Ground Available	Dozer			20
7	Load	3	Trucks	Soil StockPile	Loader		50
8	Load	4	Trucks	Soil StockPile	Loader		50
9	Haul	4					70
10	Dump	4	Loaded Truck	Load	Spotter		100
11	Return	4					120
12	Dump	5	Loaded Truck	Load	Spotter		100
13	Spread Dirt	6	Dumped Load	Dozer			150

Table 2.4 Earthmoving Activity Table

	Loop Number	Queue Node Number	Resource	Activity Node Number	Activity Description	Shared With	Shared With	Shared With
1	2	30	Dozer	20	StockPile Soil	Queue # 10 - Ground Available		
2	1	40	Soil StockPile	50	Load	Queue # 60 - Loader	Queue # 130 - Trucks	
3	3	60	Loader	50	Load	Queue # 40 - Soil StockPile	Queue # 130 - Trucks	
4	4	130	Trucks	50	Load	Queue # 40 - Soil StockPile	Queue # 60 - Loader	
5	1	80	Load	100	Dump	Queue # 90 - Loaded Truck	Queue # 110 - Spotter	
6	4	90	Loaded Truck	100	Dump	Queue # 80 - Load	Queue # 110 - Spotter	
7	5	110	Spotter	100	Dump	Queue # 80 - Load	Queue # 90 - Loaded Truck	
8	1	140	Dumped Load	150	Spread Dirt	Queue # 170 - Dozer	Queue # 90 - Loaded Truck	
9	6	170	Dozer	150	Spread Dirt	Queue # 140 - Dumped Load		
10	1	10	Ground Available	20	StockPile Soil	Queue # 30 - Dozer		

Table 2.5 Earthmoving Resource List Table

Observing Table 2.4, for each simulation loop within the model, the activities for each cyclic loop are spelled out to the user in order. Combi activities, which require resources, have the required resources listed beside them. Observing Table 2.5, each resource in the model is listed along with its activity and any associated resources required to carry out the activity. The user has now built a simulation model that has captured and spelled out the construction process used in the model, and also has a record of the resource allocations in the project.

2.9 Conclusion

The keystone to the descriptive modeling process is the structure the user must follow when modeling. If no attention is given to the loop number and element identification number system, the descriptive modeling process will not work. The modeler must identify the cyclic loops and paths within the model then determine the activities within these loops before modeling. After the creation of the model, the computer must simply read the constructed model and draw out the information required to describe it.

The method of reading the model and sorting the elements according to loop numbers and element identification numbers is the manner in which position is assigned to the elements in the simulated construction process. For example, element 10 in loop 1 comes before element 20 in loop 1 and loop 1 is a different process than loop 2. Resources for the activities are always placed before the Combi activities. It is this structure that allows the resources to be identified with their activities and the descriptions used to spell out the construction processes are parameters defined within the elements. Thus, the modeling structure is key to the performance of the descriptive process.

The output for the prototype model is the Activity Table and the Resource List Table. Both tables can be used to spell out the construction process quite clearly. The Activity table will spell out the activities, in order of construction, for each loop, and will also indicate what resources are needed for what activities. The Resource List table details the resource allocation throughout the simulated project. It indicates what resources are required for what activities.

At present, these tables are very functional in describing the construction process. One concern of the author, however, is the aesthetics of the tables. At the moment, the system describes just the static state of the model: i.e. how the user has designed the model. Further improvements to the system could involve the incorporation of dynamic aspects of the model, including statistics describing the use of the resources in the model. It could also describe how long a resource must wait at a certain queue, or what the utilization of a server is for a given resource allocation. A second improvement would be the creation of a dialogue to describe the model, rather than using the tables presented here. Rather than using tables, which the user could find confusing, the output could be in the form of English sentences that create a dialogue describing the processes occurring in the simulation model. Additional inputs in each of the elements could allow for the output dialogue to include specification data significant to the construction at hand. Thus, it is the feeling of the author that for a program such as this to be accepted into industry, customization of these tables or a dialogue for the user's needs may be required. Some users may require more detail or detail specific to their interests that would make this system a domain specific system. At this point in time, it is unknown to the author as to the specific necessities of industry, and thus the system has been kept generic.

Chapter 3

Simulation Modeling Template

3.1 Introduction

The fundamentals for this simulation template are the concepts taken from CYCLONE created by Halpin (Vanegas, Bravo, and Halpin, 1993). CYCLONE was a modeling tool that consisted of seven basic elements that when linked together effectively modeled cyclic operations. During the development of this prototype, some of the limitations of CYCLONE were encountered and therefore adaptations to the template were made.

Some of the adaptations to the modeling template were the addition of parameters required to describe the activities and resources in the model, and to calculate performance indices that were used to evaluate the resource allocation in the model so that resource allocation improvement could occur. As explained in the previous chapter, another adaptation to the CYCLONE template was constructing it with the limitation of only four connection points for each element. The last major adaptation worthy of note was the direction of entities in the simulation model. The original CYCLONE was not concerned with the correct flow of the entities though the system, as long as the entities flowed through the system. Thus, an entity that started out in one cyclic operation could end up in another, and cyclic operations started by one entity could be completed by another. The problem encountered was in the calculation of the performance factors. Cycle times and other related information were required for the calculation of these indices, which will be described in detail later in this chapter. Therefore, by directing the entities through their cyclic loops, the calculation of the performance indices was ensured.

This chapter introduces the reader to the modeling template and how it is used to create simulation models, and is broken down into the following sections: section 3.2 introduces the reader to the template modeling elements, their function, and how they are defined; section 3.3 gives an overview of the modeling approach used to create a simulation model so that the descriptive modeling process and the resource allocation improvement process function correctly; section 3.4 gives an example of creating a simulation model using the modeling elements; and section 3.5 is the conclusion of this chapter and summarizes its main points.

3.2 Overview of the Modeling Elements

There are nine modeling elements used in this modeling template: the CYCLONE Parent Element, the Server Queue Element, the Customer Queue Element, the Combi Activity Element, the Normal Activity Element, the Consolidate Element, the Generate Element, the Counter Element, and the relationship used to link the elements together. As stated before, these elements closely follow the CYCLONE modeling elements and the constraints used when modeling in CYCLONE; there are, however, some slight differences between them.

3.2.1 Parent Element

This element (Figure 3.1) is one in which the descriptive Cyclone modeling elements can be placed. Simphony allows for hierarchical modeling; in other words, Simphony allows simulation modeling to occur within modeling elements. As a means of control, the descriptive CYCLONE modeling template has been created to limit this hierarchical structure to be developed within the simulation model. This is done by allowing the simulation model to be developed only in the CYCLONE parent modeling element. Thus the descriptive modeling elements can only be placed within the Parent Element, and they can have no children.

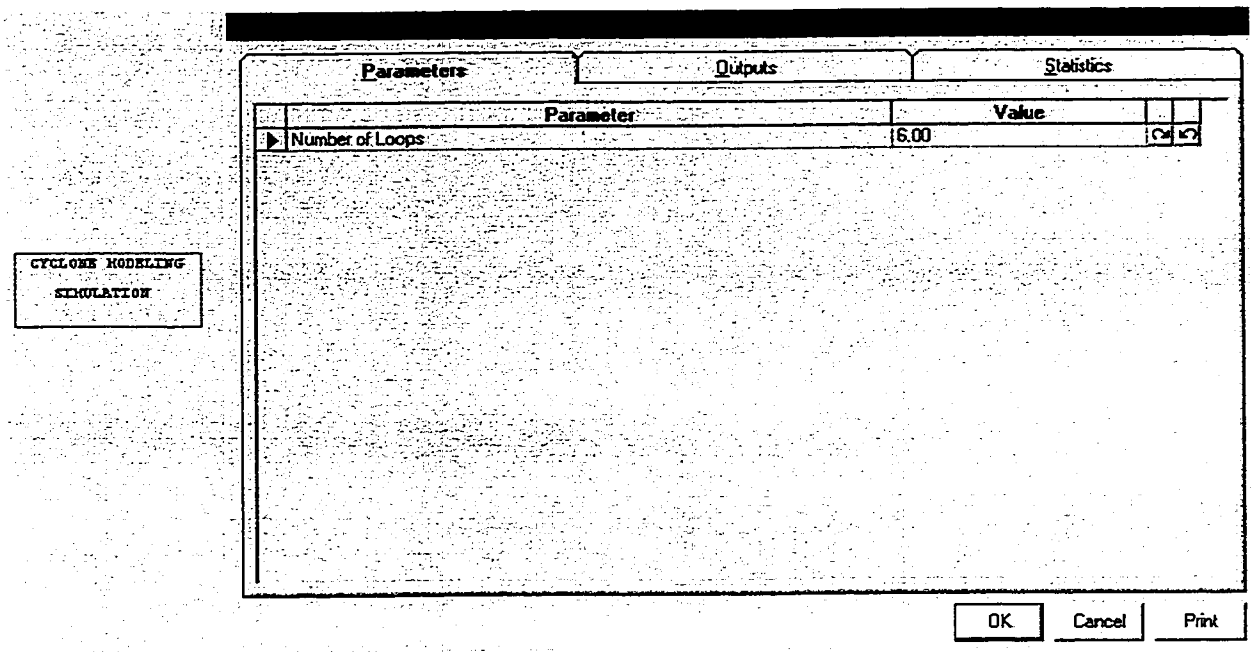


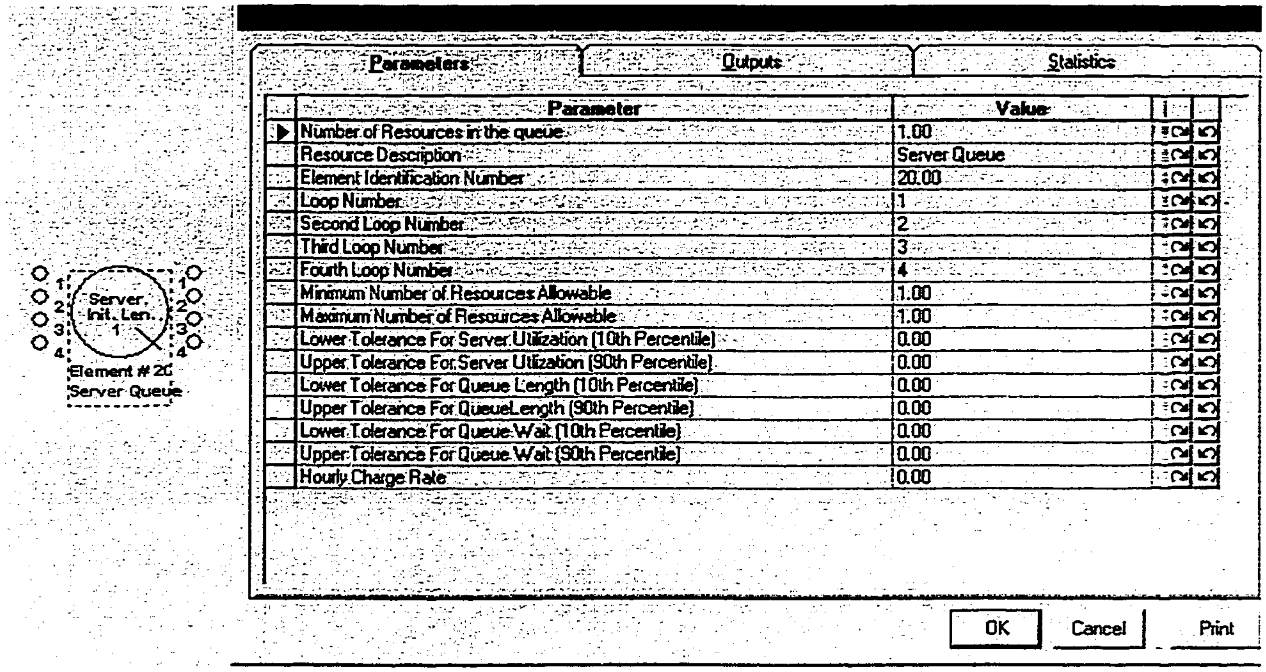
Figure 3.1 Cyclone Parent Element and Parameters

The only parameter for this element is the Number of Loops in the simulation model, and this number is used when creating the drop down list boxes for the modeling elements. Outputs of this element are the Activity Table and the Resource List Table. These tables are the outputs from the descriptive modeling process and can be found in the Output folder of the Parent Element. The Activity Table describes the activities that take place in the model and the resources required to carry out these activities. The Resource List Table is a list of the resources that are needed in the model and how they are allocated to the activities in the model. Using these two lists the construction method, the resources required, and the allocation of the resources for the model can be determined.

The hidden outputs for this element, those that the user cannot see, are the two hidden tables used in the descriptive modeling process and the total duration of the project. The total duration for the project is the total simulation time for which the project ran. This duration is used to track the variance in the project duration when running the resource allocation improvement routine in Visual Basic. The user has the option to terminate the resource allocation improvement process if either the cost or duration, depending on what is being taken into consideration, does not fluctuate over a user-defined tolerance value in the last five iterations of the resource allocation improvement process.

3.2.2 Server Queue Element

The Queue Element in Cyclone is broken into two new Queue Elements in this modeling template: the Server Queue Element and the Customer Queue Element. This division of the Queue Element is done due to the manner in which the resource allocation improvement process looks at the model as interactions between servers and customers. The terms "Server" and "Customer" apply here as they would apply in queuing theory, where a server is something that provides a service to an incoming customer. Using a building construction example, the crane onsite would be viewed as a server for the beams that require placement, which would be considered customers.



Parameter	Value
Number of Resources in the queue	1.00
Resource Description	Server Queue
Element Identification Number	20.00
Loop Number	1
Second Loop Number	2
Third Loop Number	3
Fourth Loop Number	4
Minimum Number of Resources Allowable	1.00
Maximum Number of Resources Allowable	1.00
Lower Tolerance For Server Utilization (10th Percentile)	0.00
Upper Tolerance For Server Utilization (90th Percentile)	0.00
Lower Tolerance For Queue Length (10th Percentile)	0.00
Upper Tolerance For QueueLength (90th Percentile)	0.00
Lower Tolerance For Queue Wait (10th Percentile)	0.00
Upper Tolerance For Queue Wait (90th Percentile)	0.00
Hourly Charge Rate	0.00

Figure 3.2 Server Queue Element and Parameters

Therefore the Server Queue Element (Figure 3.2) is a modeling element used to represent a sever resource such as a loader, crane, mechanic, or inspector. It acts like a CYCLONE Queue Element by creating entities and making them wait until they are called upon. As in CYCLONE, this element and the Customer Queue Element are the only elements allowed to precede Combi Elements. Inputs for this element are as follows:

Number of Resources In The Queue - This is the number of entities initially in the queue. The resource allocation improvement process will change this number as required.

Resource Description - This is a description of the resource(s), which are the entities created within the queue.

Element Identification Number - This is the number used to identify the element. As an element is created, a number is assigned to the element. The user has the ability to change this number though, since it is also used to assign positioning of the element within the model and to assign priority of entity allocation for the element during simulation.

Loop Numbers For The Connection Points - Each element, with the exception of the Parent Element, the Consolidate Element, and the Generate Element, has four incoming and outgoing connection points from which to connect to other elements. Each connection point is only allowed to have one Relationship associated with it. The loop number assigned to the connection point identifies which loop the entity is in. The element identification number identifies the element's position within the loop. Each element is created with one set of connection points. Assigning loop numbers to any of the Loop Number parameters can attain additional connection points. A value of "N/A" for a loop number parameter indicates that no connection point is required.

Minimum Number of Resources Allowable - This is the minimum number of resources allowed for the model. It is a constraint that the user defines and that the resource allocation improvement process must follow. The resource allocation improvement process cannot allocate fewer resources to the Server Queue Element than specified by this parameter.

Maximum Number of Resources Allowable - This is the maximum number of resources allowed for the model. It is a constraint that the user defines and which the resource allocation improvement process must follow. The resource allocation improvement process cannot allocate more resources to the Server Queue Element than the number specified by this parameter.

Lower Tolerance for the Server Utilization - This is the lowest tolerance that the user specifies for the server utilization. It is used as a performance measure such that recommendations can be made to the model during the resource allocation improvement process.

Upper Tolerance for the Server Utilization - This is the highest tolerance that the user specifies for the server utilization. It is used as a performance measure such that recommendations can be made to the model during the resource allocation improvement process.

Lower Tolerance for the Queue Wait - This is the lowest tolerance that the user specifies for the Average Queue Wait Time. It is used as a performance measure such that recommendations can be made to the model during the resource allocation improvement process.

Upper Tolerance for the Queue Wait - This is the highest tolerance that the user specifies for the Average Queue Wait Time. It is used as a performance measure such that recommendations can be made to the model during the resource allocation improvement process.

Lower Tolerance for the Queue Length - This is the lowest tolerance that the user specifies for the Average Queue Length. It is used as a performance measure such that recommendations can be made to the model during the resource allocation improvement process.

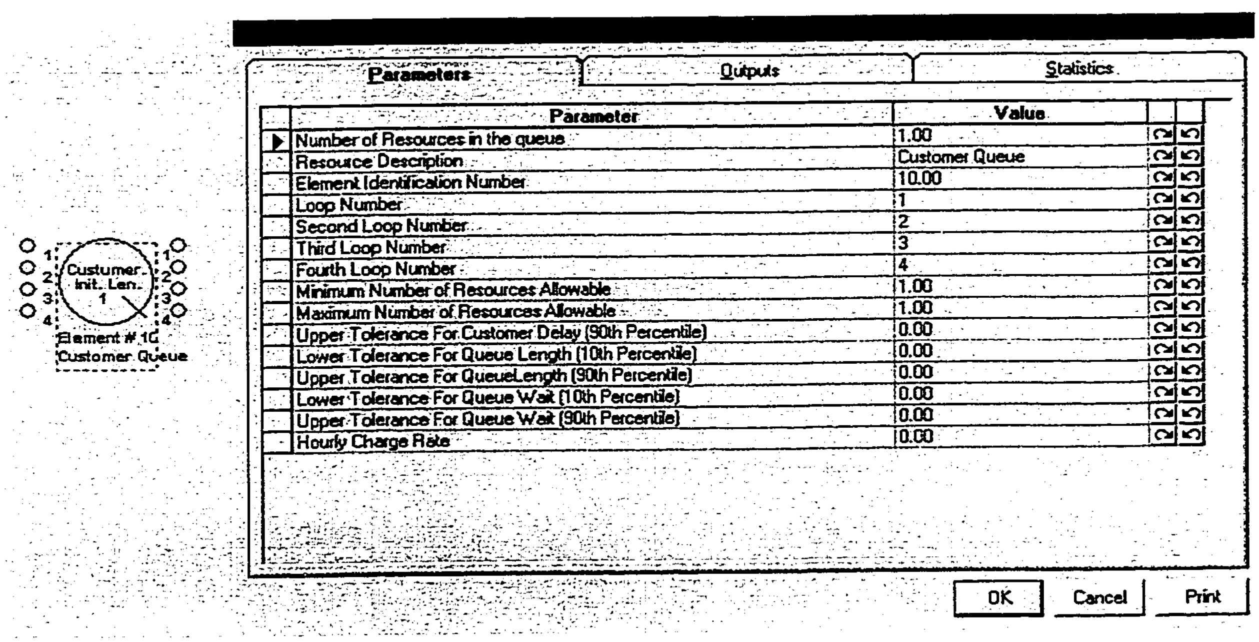
Upper Tolerance for the Queue Length - This is the highest tolerance that the user specifies for the Average Queue Length. It is used as a performance measure such that recommendations can be made to the model during the resource allocation improvement process.

Hourly Charge Rate - This is the hourly charge rate for a given resource.

Outputs from this model are the resource Start Time, resource Finish Time, the Total Delay for the resource(s), the productive Time Of Use for the resources(s), the Server Utilization, the Average Queue Wait, and the Average Queue Length. Hidden outputs for this element are the total cost for the element, tolerances for server utilization, the tolerances for average queue length, the tolerances for average queue wait, and whether or not the Server Queue element is allowed to increase and decrease its resource numbers. The cost of the Queue element is determined by multiplying the number of resources by the time they were used in the project and again by their hourly cost.

3.2.3 Customer Queue Element

The Customer Queue (Figure 3.3) is quite similar to the Server Queue. The only difference between the two elements is that, where the Server Queue was concerned with the server utilization, the customer queue is concerned with the Customer Delay Index. Thus, all other input parameters for this queue are the same as that of the Server Queue, with the exception of the server utilization and its tolerance parameters; the customer delay index and its upper tolerance replace them.



Parameter	Value
Number of Resources in the queue	1.00
Resource Description	Customer Queue
Element Identification Number	10.00
Loop Number	1
Second Loop Number	2
Third Loop Number	3
Fourth Loop Number	4
Minimum Number of Resources Allowable	1.00
Maximum Number of Resources Allowable	1.00
Upper Tolerance For Customer Delay (90th Percentile)	0.00
Lower Tolerance For Queue Length (10th Percentile)	0.00
Upper Tolerance For QueueLength (90th Percentile)	0.00
Lower Tolerance For Queue Wait (10th Percentile)	0.00
Upper Tolerance For Queue Wait (90th Percentile)	0.00
Hourly Charge Rate	0.00

Figure 3.3 Customer Queue Element and Parameters

The Outputs from this element are the Number Of Cycles completed, the sum total of the cycle times, the sum total of the delays experienced during the cycles, the average cycle time, the average time of delay experienced during a cycle, the average Customer Delay

Index, the Average File Length, and the Average Queue Length. The hidden outputs for this element are the total cost for the element, upper tolerance for the Customer Delay Index, the upper and lower tolerances for Average Queue Length and Average Queue Wait, and whether or not the Customer Queue element is allowed to increase and decrease its resource numbers. The cost of the Queue element is determined in the same manner as the Server Queue element, by multiplying the number of resources by the time they were used in the project by their hourly cost.

3.2.4 Combi Activity Element

Activities within the model can occur in two modeling elements, the Combi Element and the Normal Element. These elements are very similar, the only difference being that the Combi Element requires a queue element before it. Thus, the Combi Element (Figure 3.4) is where an activity requiring one or more resources is modeled. It simulates an activity by causing an entity or entities to experience a time delay. In the case of multiple queue elements being linked to the Combi Element, each queue element must be able to provide the Combi Element with an entity for its activity to occur. If each queue element cannot supply the Combi Element with an entity, these entities will remain at the queue elements until an entity can be provided. The outputs for the Combi element are the Number of Served Entities and the Activity Service Time. Inputs for this element are as follows:

Activity Service Time - The Activity service time is done using the CFCSIM_Sampler class. It allows the generation of random variants based on a constant, beta, exponential, normal, triangular, or uniform distribution. These variants are used to assign a duration to the activity that the element is to represent.

Activity Description - This is the description of the activity that is to take place. The attribute for this parameter can be defined using phrases such as "Load Trucks" or "Place Beam".

Element Identification Number - This parameter is the same as that described for the Server Queue Element in section 3.2.2.

Loop Numbers for the Connection Points - This parameter is the same as that described for the Server Queue Element in section 3.2.2.

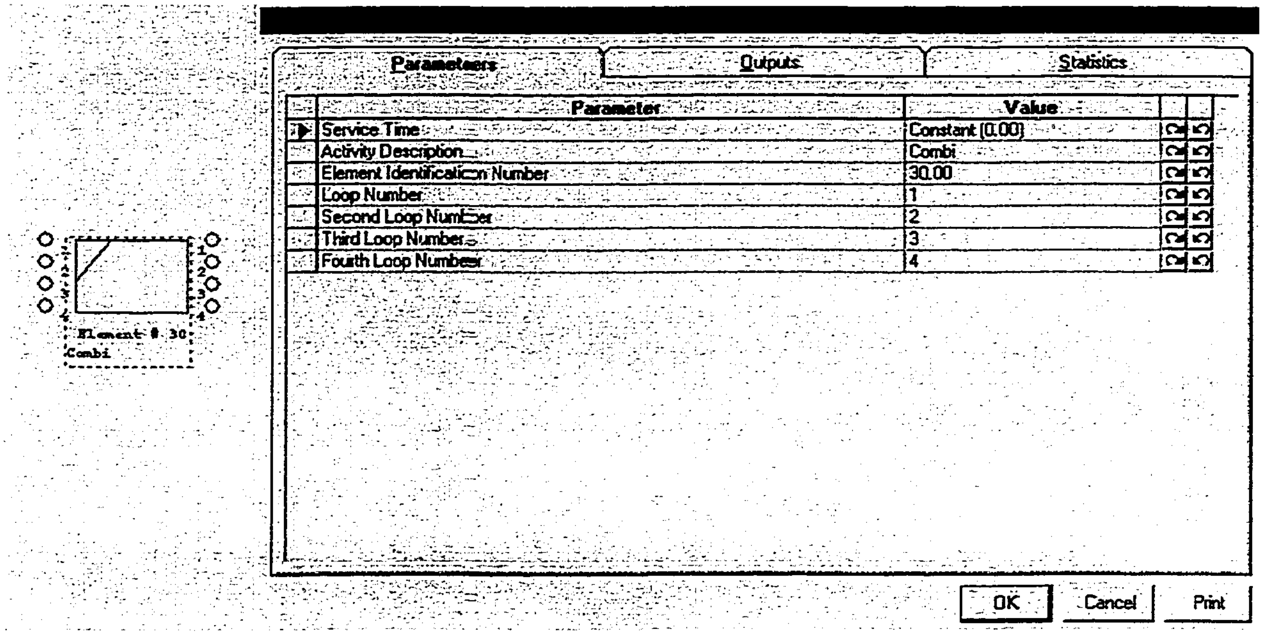


Figure 3.4 Combi Element and Parameters

3.2.5 Normal Activity Element

For the most part, the Normal Element (Figure 3.5) is the same as the Combi Element, the only difference being the lack of need for a queue element before it. When an entity arrives at the Normal Element, it is simply delayed to simulate the activity duration. Unlike the Combi Element, the Normal Element does not need all its preceding elements to be able to provide it with an entity for its activity to occur, as it just processes the entities as they arrive.

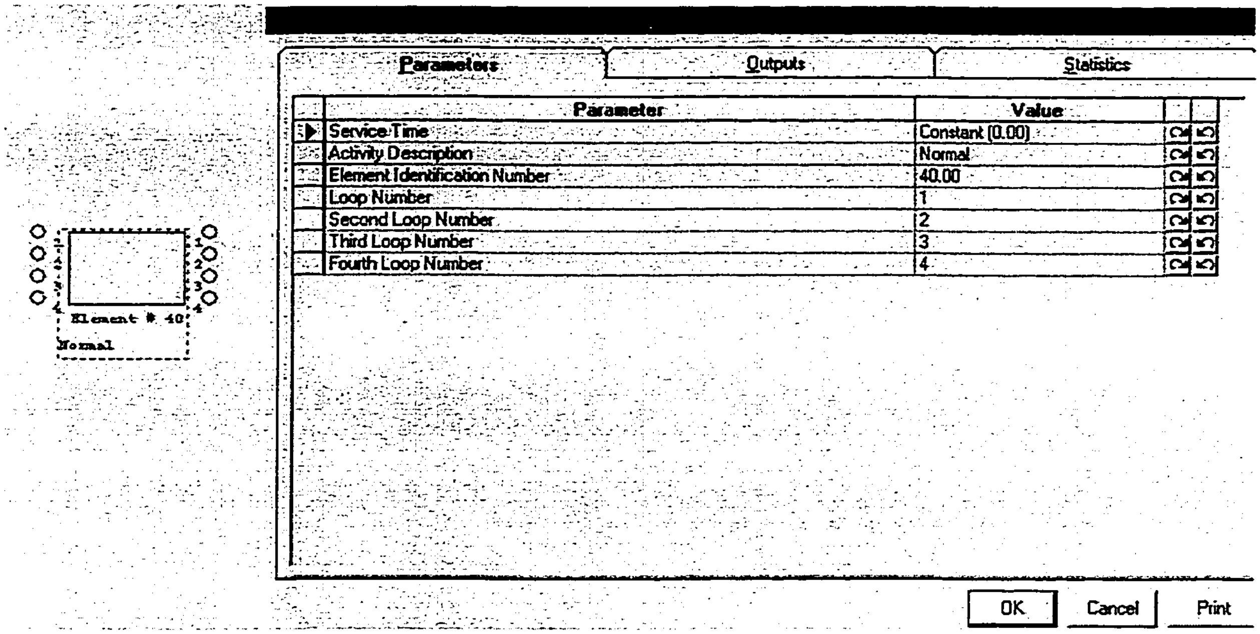


Figure 3.5 Normal Element and Parameters

3.2.6 Consolidate Element

The Consolidate Element (Figure 3.6) is used to combine several entities into one. This element has only one incoming and outgoing connection point, and the entity that leaves the element assumes the properties of the last entity to come into the element. Inputs for this element are as follows:

Number to Consolidate - This threshold value is the number of entities to combine into one entity. For example, if the threshold value is five, five entities will come into the entity and only one will leave.

Element Identification Number - This parameter is the same as that described in the Server Queue Element in section 3.2.2.

Loop Numbers for the Connection Points - This parameter is the same as that described in the Server Queue Element in section 3.2.2.

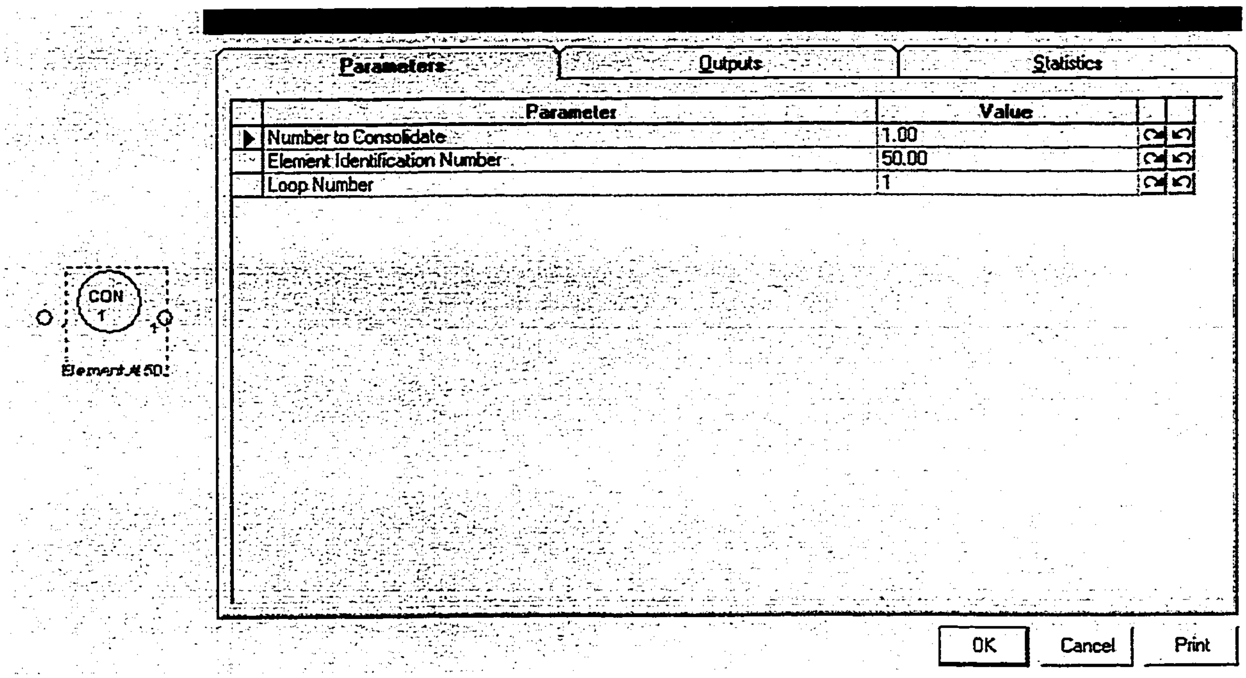


Figure 3.6 Consolidate Element and Parameters

The only output for this element is the number of entities sent out of the element.

3.2.7 Generate Element

The Generate Element (Figure 3.7) is used to clone incoming entities. This element has only one input and one output connection point. Each entity that is cloned retains the properties of the incoming element. Inputs for this element are as follows:

Number to Generate - The number to generate is the number of entities to clone each time an entity arrives in the element. For example, if the Number To Generate is 5 and one entity comes into the Generate Element, five leave.

Element Identification Number - This parameter is the same as that described in the Server Queue Element in section 3.2.2.

Loop Number for the Connection Point - This parameter is the same as that described in the Server Queue Element in section 3.2.2.

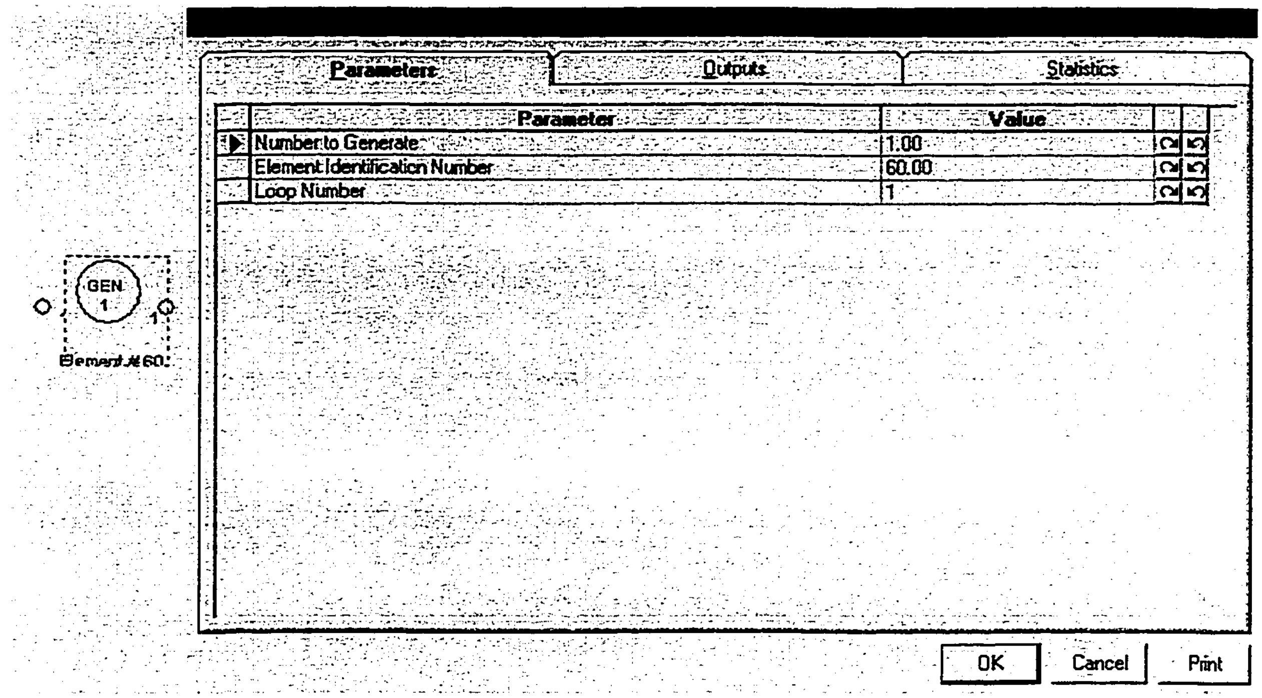


Figure 3.7 Generate Element and Parameters

The only output for this element is the number of entities sent out of the element.

3.2.8 Counter Element

The Counter Element (Figure 3.8) is a statistical element used to count the number of entities that pass through it. It can be used to terminate a simulation by specifying a tolerance number of counted entities after which to stop the simulation run. Its inputs are as follows:

Multiplier - This number multiplies the number of entities that are counted by a user-defined value. Thus, if the multiplier is 5 and 4 entities have passed through the counter, the total quantity of entities that have been counted is 20.

Target Quantity To Stop Simulation - This is the user-defined variable that can be used to terminate a simulation run.

Element Identification Number - This parameter is the same as that described in the Server Queue Element in section 3.2.2.

Loop Number for the Connection Points - This parameter is the same as that described in the Server Queue Element in section 3.2.2.

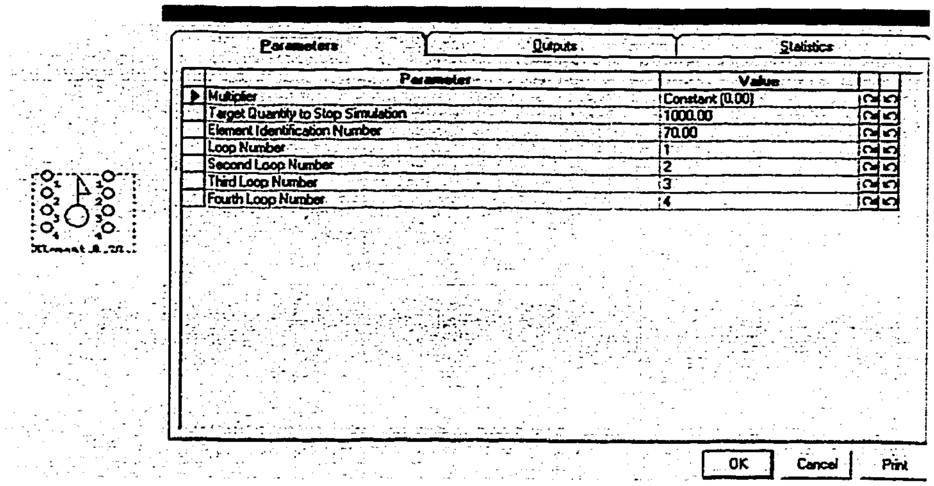


Figure 3.8 Counter Element and Parameters

3.2.9 Relation

The Relation (Figure 3.9) is used to link two elements together, allowing entities to travel between the modeling elements. Travel along the Relation is in one direction only; the entity travels from the source object to the destination object linked by the Relation. In this template, only one Relation is allowed per connection point. The Relation has no input or output parameters.

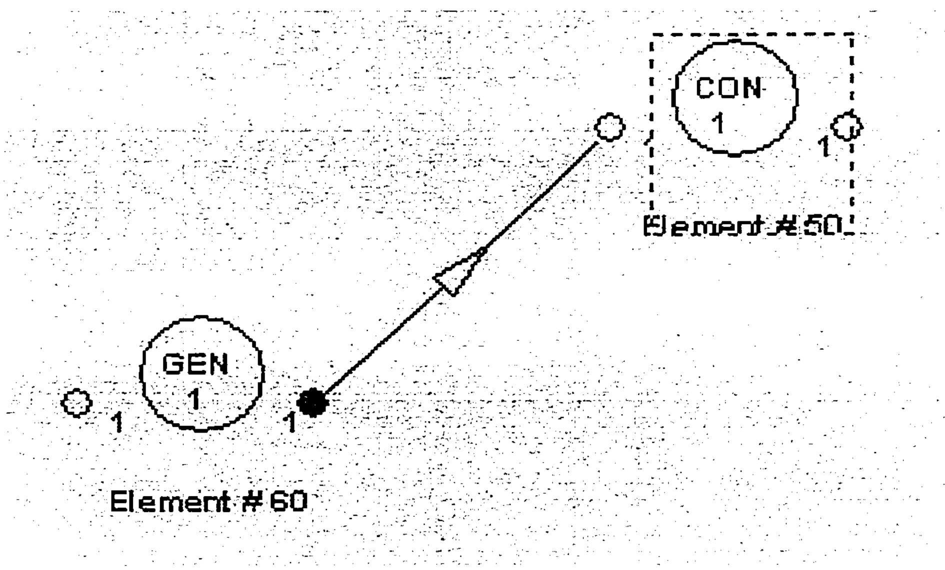


Figure 3.9 Relation

3.3 Overview of Modeling Approach

There are two considerations to keep in mind when modeling with this template: the descriptive modeling process and the resource allocation improvement process (Figure

3.10). The descriptive modeling process works on the loop and element identification number structure described in the previous chapter. The other concern is the resource allocation improvement modeling process, which works on cyclic loops that use cycle times and the loop's queuing characteristics to calculate performance indices. These performance indices, created during the simulation of the model, describe the current state of the customer/server relationships within the model, and are used as inputs into the Belief Network. The Belief Network uses these indices to evaluate the current state of the model and to make recommendations to improve the model's performance. Once the Belief Network has compiled a list of possible recommendations, the Visual Basic interface evaluates the recommendations and presents the user with a list of viable recommendations. These recommendations are ranked by their associated belief values. From this list one recommendation is chosen with which to improve the simulation model.

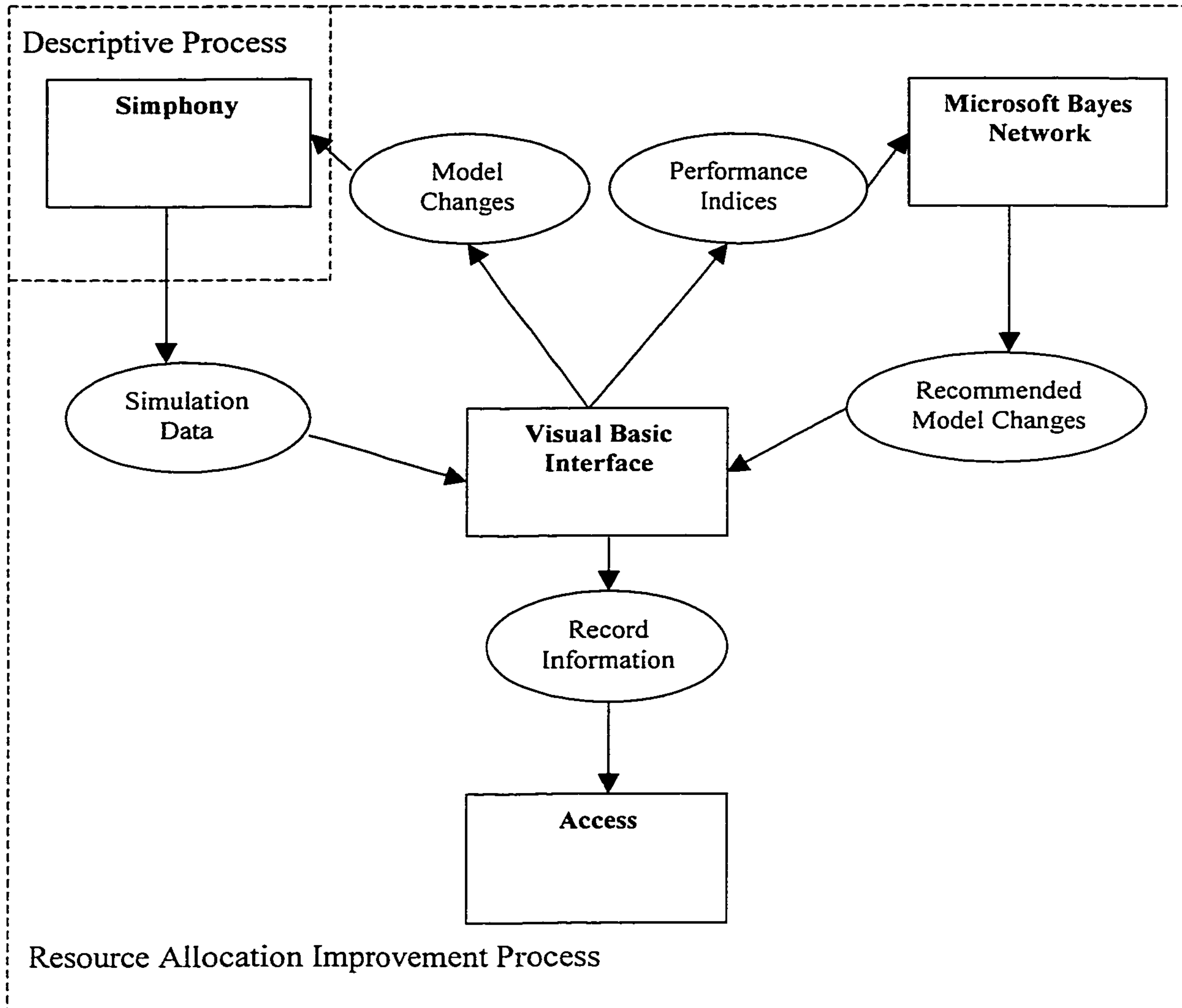


Figure 3.10 System Overview

3.3.1 Overview of Descriptive Modeling

As explained in the previous chapter, the descriptive modeling process works on a loop and element identification number system. The loop numbers identify the different processes within the simulation model. The element identification numbers identify an element's position within these loops.

Essentially, this process requires the user to model the system so that it follows this loop and element identification number structure. To do this, the user must first look at the system they wish to model and identify the individual processes within it. These individual processes make up the loops for the simulation model; they are the paths that entities follow. Examples of these loops are a truck route in an earthmoving model, or a labor crew routine on a repetitive construction project. The second task the modeler must perform is to identify the order of the activities in these loops and assign their element identification numbers accordingly. The activities should be placed within their loops with their activity numbers ranging from lowest to highest according to their order in the construction method. Once the modeler has built the model using these constraints, when the model is simulating, the computer will search the model and record the information describing it in the Activity Table and Resource List Table, located in the output folder of the Parent Element.

3.3.2 Descriptive Modeling Parameters

The most important parameters for the descriptive modeling process are the element identification numbers and the loop element numbers. These parameters give structure to the model. The loop numbers define the different processes within the model and the element identification numbers identify the modeling element positions within these loops. Upon creation, each element is assigned an element identification number that the user has the ability to change. The first element created receives an element identification number of 10, and each additional element created receives an element identification number incremented by 10. The loop numbers for the elements are defined by selecting the appropriate loop number(s) from the drop-down list boxes for each of the elements four connection points. All elements, with the exception of the Parent, Consolidate, and Generate Elements, have four connection points. The value in the Parent Element's Loop Number parameter determines the number of loops listed in the drop-down list boxes for the modeling elements. Unless otherwise specified all loop number parameter list boxes, with the exception of the first one, have been set to "N/A". This indicates that there are no additional connection points needed for other loops.

Information relevant to describing the processes within the loops is defined in the parameters of the Customer Queue Element, Server Queue Element, Combi Element, and Normal Element. The Combi and Normal Elements contain information regarding the activities occurring within the model. The information required from these elements is the phrase that describes the activity occurring, found in the Activity Description parameter of the Normal and Combi Elements. Information regarding the resources used in the model is found in the Customer Queue and Server Queue Elements. The information needed from these elements is the name of the resource that is stored within them. This phrase is found in the Resource Description parameter of the Server Queue and Customer Queue Elements.

3.3.3 Overview to Resource Allocation Improvement Modeling

The resource allocation improvement modeling process works by attaining information from the simulation model so that its state can be determined and recommendations can be made to improve the model. For this prototype, the method used to gain information about the state of the model is through performance indices. Microsoft Bayes Network is used to create a belief network that uses these indices to come up with model improvement recommendations. The tool that works to tie the simulation model and belief network together is the Visual Basic interface. This interface also evaluates the recommendations generated in the belief network and selects the appropriate recommendation to apply to the model.

The resource allocation improvement portion of this prototype has been developed for construction projects with cyclic activities, due to the manner in which information is collected to calculate the performance indices. The information for these indices is collected from the entities at the queue modeling elements in which they were created. Thus, it should be apparent why the need for the direction of the entities and the cyclic nature of the prototype is required: both the Customer and Server Queue Elements act as check points where entities record their cycle times and delay times. This process is accomplished by assigning a name to the entities created within the Queue. This name is composed of the queue element's type and element identification number. In the case of entities created in the Server Queue # 10, the entities created within that queue are all named "Server10". Thus, every time an entity named "Server10" comes into the Server

Queue # 10, the Server Queue Element knows to record the active time of the entity such that the server utilization can be calculated. In the case of a Customer Element, every time a entity named "Customer20" comes into the Customer Queue # 20, the Customer Queue Element knows to collect cycle time and entity delay data such that the Customer Delay Index can be calculated. The Average Queue Wait and Average Queue Length outputs are intrinsically generated statistics from the Simphony file in which the waiting entities are stored within the queue.

The method of collecting data for generating indices explained here is for cyclic activities, but it could easily be applied to non-repetitive processes. This problem is outside the scope of this thesis, but it could be easily be solved by creating two additional modeling elements. One element would generate and label the entities at the beginning of the entity's path, while the other element would be placed at the end of the entity's path and would be used to collect information from the entity and generate the required performance indices.

3.3.4 Resource Allocation Improvement Parameters

The parameters required for the resource allocation improvement process are used to calculate and then measure the performance indices. The indices used to measure the performance of this model are the Average Queue Wait, Average Queue Length, Customer Delay Index, and Server Utilization. The tolerance parameters, which the user defines, are the acceptable limits between which these performance indices can occur.

All calculations of the indices occur at the queue modeling elements. The Customer Queue calculates the Customer Delay Index, Average Queue Length, and Average Queue Wait. The Server Queue calculates the Server Utilization, Average Queue Length, and Average Queue Wait. It should be noted that only one belief network is required for the resource allocation improvement process and it does not need all the performance indices to formulate a recommendation.

Additional parameters used in the resource allocation improvement process limit the number of resources that can be used. The user is allowed to specify a minimum and maximum number of resources to be used in a project. The resource allocation improvement algorithm must respect these limits and will not assign a resource number outside of the specified limits.

3.3.2 Performance Indices

Performance indices are calculated numbers that describe the states of the Server and Customer Queues during simulation. The indices used to describe the state of the Server Queue modeling element are the Server Utilization, Average Queue Length, and Average Queue Wait. The indices used to describe the state of the Customer Queue modeling element are the Customer Delay Index, the Average Queue Length, and the Average Queue Wait.

Upon the completion of a simulation run, the calculation of these indices is performed for each queue element. The tolerance values used to evaluate the state of the performance indices are user-defined. The Server Utilization, Average Queue Length, and Average Queue Wait have upper and lower tolerances, while the Customer Delay Index has only an upper tolerance limit. Due to the fact that simulation is based on a state of randomness, there has been a margin of safety recommended for the user-defined tolerance limits. The user is asked to input the 10^{th} and 90^{th} percentile values for the upper and lower tolerance limits respectively. Because the simulation process has a random characteristic associated with it, there is a chance that the actual construction may be carried out at a rate faster or slower than simulated. Thus, by using a more constrictive tolerance range, there is insurance that the impact from the deviation of actual construction from the simulated construction will have minimal impact.

The chapters to follow will give the reader a fuller understanding of the resource allocation improvement process and the function the performance indices serve. For the moment, a brief definition will be given for each of the performance indices. The Average Queue Wait is the average time an entity waits between the time it enters a queue element and when it is used by a Combi Element. The Average Queue Length is the average length of the queue being the number of entities waiting in the queue. The Server Utilization is the ratio of the server's productive time to the total time the server is used. The Customer Delay Index is the ratio of the total delay time an entity experiences as it travels through its cyclic operation compared to its average cycle time.

3.4 Creating a Simulation Model

McCabe (1997) used a queuing model created by Carmicheal (1987) to verify her automated modeling approach for construction performance improvement using computer simulation and belief networks. The model described a truck and shovel operation, and the number of trucks in the system was improved with respect to cost. McCabe employed a similar method of using performance indices to describe the state of the model to the belief network. Thus, this queuing model will also be used in this thesis to validate the resource allocation improvement process, and will be discussed in chapter 6.

The model consists of one shovel and several trucks. The number of trucks is allowed to vary between two and ten. The average arrival rate of the trucks is 4.5 trucks per hour, and the average service rate of the trucks was 31.1 trucks per hour. Both of these activities are modeled using exponential distributions. The cost ratio of shovel to truck is 2 to 1. The upper and lower limits for the server utilization are 0.25 and 0.95 respectively. The upper limit for the customer delay index is set at 0.3. The limits for the Average Queue Length and Average Queue wait are dependent on the number of customers in the system and can be found in Table 3.1. The simulation is to run for 100 truck cycles.

Number Of Customers	2	3	4	5	6	7	8	9	10
Lower Wait Limit	0.019	0.064	0.139	0.251	0.410	0.618	0.882	1.201	1.569
Upper Wait Limit	0.141	0.432	0.857	1.389	2.019	2.725	3.497	4.315	5.160
Lower Length Limit	0.094	0.307	0.664	1.180	1.880	2.762	3.822	5.038	6.370
Upper Length Limit	0.243	0.735	1.440	2.311	3.333	4.471	5.709	7.019	8.367

Table 3.1 Upper and Lower Tolerance Limits for Average Queue Wait and Length

The first step in modeling the truck and shovel operation is to identify the loop and element identification structure in the model. In this model there are two loops: the truck loop and the loader loop. The next step is to identify the activities within the loops. For the Loader loop, the only activity is the loading of the trucks. For the truck loop, the activities are the loading of the trucks and the activity used to represent the inter-arrival time for the trucks. The truck loop will require a counter element to terminate the simulation after 100 truck cycles.

The second step is to plot out the model. Starting with the truck loop, elements are placed in the order described with relations connecting them together with the model in Figure 3.11 as result. A Combi Element is placed for the truck loading activity. A Normal Element is placed to represent the inter-arrival time of the trucks. A Counter Element is placed to terminate the simulation once 100 cycles are complete. Lastly, a Customer Queue Element is placed to create the trucks. This element is attached to the load trucks Combi Element. The loader cycle is constructed by attaching a Server Queue Element to the truck loading Combi Element.

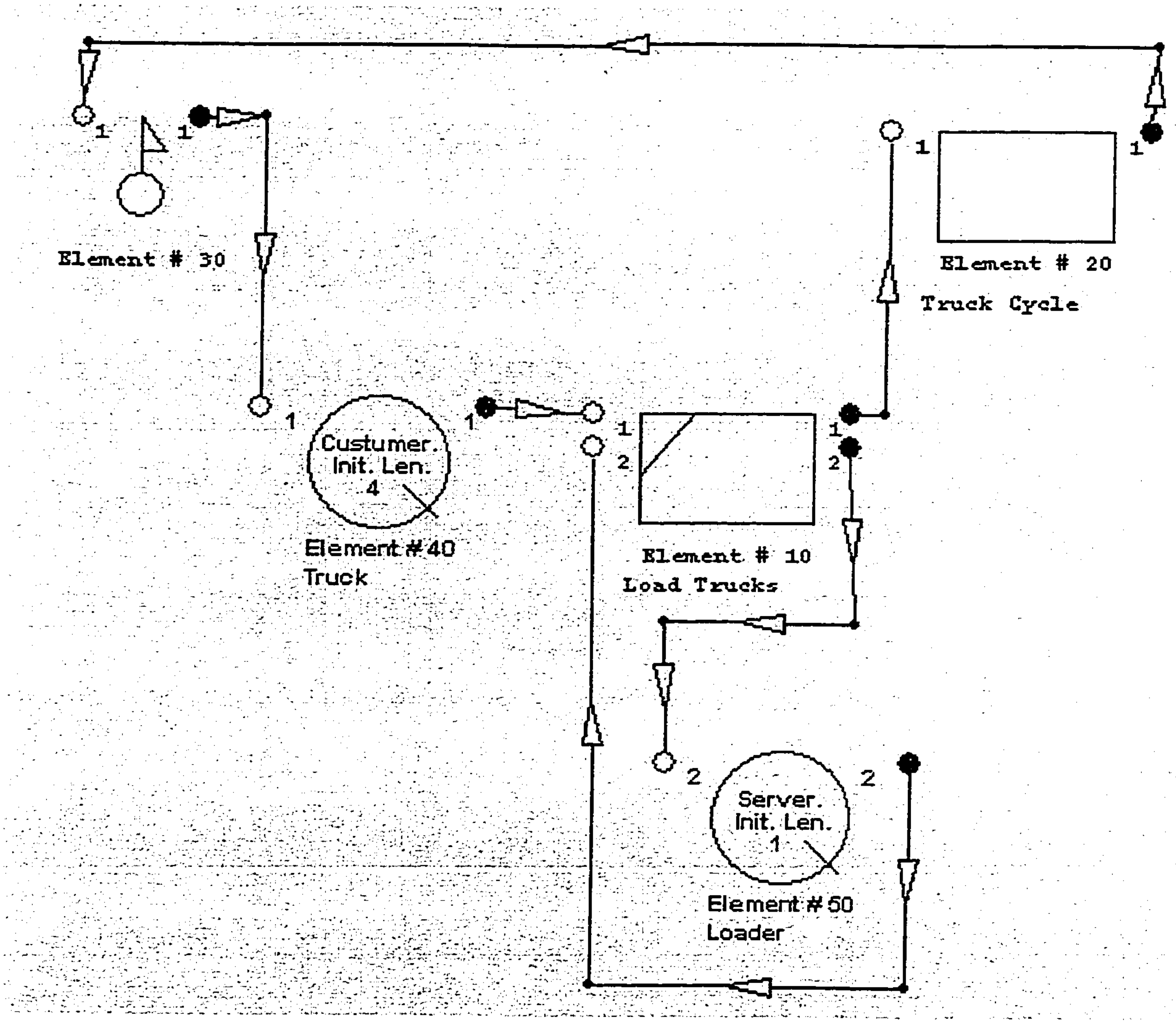


Figure 3.11 Earthmoving Simulation Model

The third and final step to creating the model is to define the attributes for the modeling elements. Using the information provided above, the parameters for the elements are defined as follows:

Element # 10 Load Trucks Combi Element

Service Time = Exponential (0.03)

- 1/31.1 trucks per hour = 0.03 hours per truck

Activity Description = Load Trucks

Element Identification Number = 10.00

Loop Number = 1

Second Loop Number = 2

Third Loop Number = N/A

Fourth Loop Number = N/A

Element # 20 Truck Inter-arrival Normal Element

Service Time = Exponential (0.22)

- 1/4.5 trucks per hour = 0.22 hours per truck

Activity Description = Truck Cycle

Element Identification Number = 20.00

Loop Number = 1

Second Loop Number = N/A

Third Loop Number = N/A

Fourth Loop Number = N/A

Element # 30 Counter Element

Multiplier = Constant (1.00)

Target Quantity to Stop Simulation = 101.00 (This allows 100 cycles to complete)

Element Identification Number = 30.00

Loop Number = 1

Second Loop Number = N/A

Third Loop Number = N/A

Fourth Loop Number = N/A

Element # 40 Truck Customer Queue Element

The definition of the Average Queue Wait and Average Queue Length parameters for this element vary with the number of trucks in the system. Thus, these parameters are dependent on the value of the Number of Resources in the Queue parameter value (Variable Name - "NumInQueue"). To create this dependency, the user must link the Average Queue Wait and Average Queue Length tolerance parameters to functions. Clicking the link button next to the parameter, as shown in Figure 3.12, opens a window in which the user can develop a function.

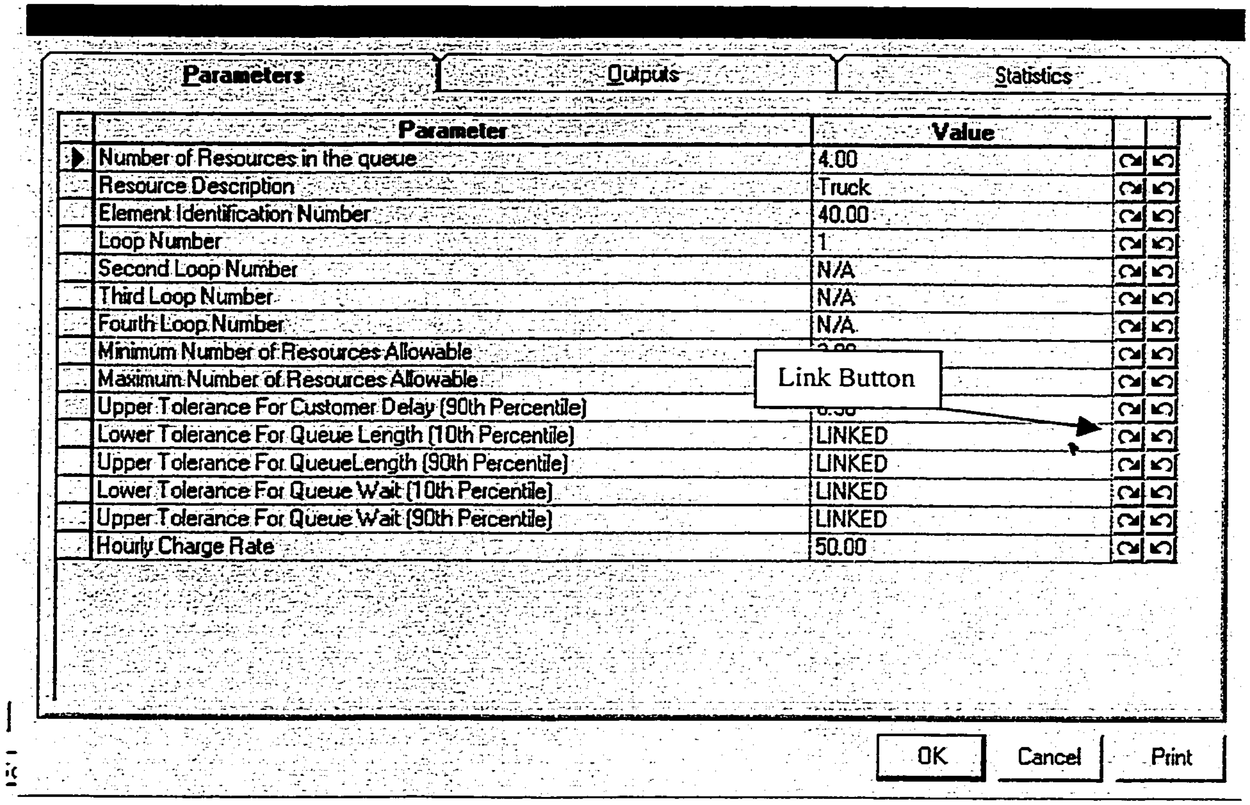


Figure 3.12 Linking a Parameter

Number of Resources in the Queue = 4 (initial value)

Resource Description = Truck

Element Identification Number = 40.00

Loop Number = 1

Second Loop Number = N/A

Third Loop Number = N/A

Fourth Loop Number = N/A

Minimum Number of Resources Allowable = 2.00

Maximum Number of Resources Allowable = 10.00

Upper Tolerance For Customer Delay Index (90^{th} Percentile) = 0.30

Lower Tolerance For Queue Length (10^{th} Percentile) = Linked

```
Public Function F289(ob As CFCSim_ModelingElementInstance)
          If ob.Attr("NumInQueue") = 2 Then
                    F289 = .094
          End If
          If ob.Attr("NumInQueue") = 3 Then
                    F289 = .307
End If
          If ob.Attr("NumInQueue") = 4 Then
                    F289 = 0.664
End If
          If ob.Attr("NumInQueue") = 5 Then
                    F289 = 1.180
End If
          If ob.Attr("NumInQueue") = 6 Then
                    F289 = 1.880
End If
          If ob.Attr("NumInQueue") = 7 Then
                    F289 = 2.762
          End If
          If ob.Attr("NumInQueue") = 8 Then
                    F289 = 3.822
End If
          If ob.Attr("NumInQueue") = 9 Then
                    F289 = 5.038
End If
          If ob.Attr("NumInQueue") = 10 Then
                    F289 = 6.370
End If
End Function
```

Upper Tolerance For Queue Length (90^{th} Percentile) = Linked

```
Public Function F290(ob As CFCSim_ModelingElementInstance)
If ob.Attr("NumInQueue") = 2 Then
                    F290 = .243
          End If
          If ob.Attr("NumInQueue") = 3 Then
                    F290 = .735
          End If
          If ob.Attr("NumInQueue") = 4 Then
                    F290 = 1.440
End If
          If ob.Attr("NumInQueue") = 5 Then
                    F290 = 2.311
End If
If ob.Attr("NumInQueue") = 6 Then
                    F290 = 3.333
End If
          If ob.Attr("NumInQueue") = 7 Then
                    F290 = 4.471
End If
          If ob.Attr("NumInQueue") = 8 Then
```

```
                F290 = 5.709
End If
        If ob.Attr("NumInQueue") = 9 Then
                F290 = 7.019
        End If
        If ob.Attr("NumInQueue") = 10 Then
                F290 = 8.367
End If
End Function
```

Lower Tolerance For Queue Wait (10th Percentile) = Linked

```
Public Function F291(ob As CFCSim_ModelingElementInstance)

If ob.Attr("NumInQueue") = 2 Then
                F291 = .000317
        End If
        If ob.Attr("NumInQueue") = 3 Then
                F291 = .001067
End If
        If ob.Attr("NumInQueue") = 4 Then
                F291 = .0023
End If
        If ob.Attr("NumInQueue") = 5 Then
                F291 = 0.004183
End If
If ob.Attr("NumInQueue") = 6 Then
                F291 = 0.006833
End If
        If ob.Attr("NumInQueue") = 7 Then
                F291 = 0.0103
End If
If ob.Attr("NumInQueue") = 8 Then
                F291 = 0.0147
End If
        If ob.Attr("NumInQueue") = 9 Then
                F291 = 0.020017
        End If
        If ob.Attr("NumInQueue") = 10 Then
                F291 = 0.02615
End If
End Function
```

Upper Tolerance For Queue Wait (90th Percentile) = Linked

```
Public Function F292(ob As CFCSim_ModelingElementInstance)
If ob.Attr("NumInQueue") = 2 Then
                F292 = .00235
        End If
        If ob.Attr("NumInQueue") = 3 Then
                F292 = .0072
End If
```

```
If ob.Attr("NumInQueue") = 4 Then
                F292 = .012483
End If
        If ob.Attr("NumInQueue") = 5 Then
                F292 = 0.02315
        End If
        If ob.Attr("NumInQueue") = 6 Then
                F292 = 0.03365
End If
        If ob.Attr("NumInQueue") = 7 Then
                F292 = 0.045417
End If
        If ob.Attr("NumInQueue") = 8 Then
                F292 = 0.058283
End If
        If ob.Attr("NumInQueue") = 9 Then
                F292 = 0.071917
End If
        If ob.Attr("NumInQueue") = 10 Then
                F292 = 0.086
End If
End Function
```

Hourly Charge Rate = 60.00

Element # 50 Loader Server Queue Element

Number of Resources in the Queue = 1

Resource Description = Loader

Element Identification Number = 50.00

Loop Number = 2

Second Loop Number = N/A

Third Loop Number = N/A

Fourth Loop Number = N/A

Minimum Number of Resources Allowable = 1.00

Maximum Number of Resources Allowable = 1.00

Lower Tolerance For Server Utilization (10^{th} Percentile) = 0.25

Upper Tolerance For Server Utilization (90^{th} Percentile) = 0.95

Lower Tolerance For Queue Length (10^{th} Percentile) = 0.00

Upper Tolerance For Queue Length (90^{th} Percentile) = 1.00

Lower Tolerance For Queue Wait (10^{th} Percentile) = 0.00

Upper Tolerance For Queue Wait (90^{th} Percentile) = 5.00

Hourly Charge Rate = 120.00

The Average Queue Length and Average Queue Wait for the Loader are not of concern, therefore their lower limits are set at zero while their upper limits are set such that they will not be exceeded.

3.5 Conclusion

This chapter introduced the reader to the modeling template and how its modeling elements are to be used in modeling a system for simulation. The basis for the template was Halpin's CYCLONE Modeling tool. Due to the limitations experienced in CYCLONE, mainly in the resource allocation improvement routine, adaptations were made in this template that deviated from the modeling methods used in CYCLONE.

This template was made up of nine modeling elements: the Parent Element, the Server Queue Element, the Customer Queue Element, the Combi Element, the Normal Element, the Consolidate Element, the Generate Element, the Counter Element, and the Relation. The function and definition of these nine elements were presented in this chapter. It was explained how to model cyclic operations using these simple elements, and an example was given. Explanation was given for modeling in regard to the two operations occurring in the in the template, the descriptive modeling process and the resource allocation improvement process. The descriptive modeling process involved modeling using a loop and element identification number system to describe the locations of the elements within the simulation model. The resource allocation improvement process involved ensuring that cyclic loops were in the model to allow the calculation of the performance indices. The cyclic loops are needed to allow the entities within the simulation model to complete cyclic operations and be able to provide data for the calculation of the performance indices, which occurs at the queue modeling element in which the element was created.

There are two recommendations for the modeling template that can be identified at this time. The first of these is the development of additional modeling elements to allow modeling of linear processes for the resource allocation improvement process. This is outside of the scope of this research, but the author sees the solution to this problem as being the creation of a Create Element where the entities are created and a Read Element where the entity information is read and the performance indices are calculated. This would require that a method be devised though to alert the Read Element that an incoming entity is from a certain Create Element for its information to be properly recorded. The second recommendation is to refine the model such that only one type of a Queue modeling element is required, and that the queuing properties are evaluated at the Combi Element. This could solve the foreseen problem of an entity being both a server and a customer. For example, a loader is a sever for the loading of trucks, but a customer for a mechanic performing maintenance. This refinement might involve generalizing the customer delay index and server utilization index into one main index. This refinement would also provide greater ease in the formulation of a simulation model with fewer modeling elements with which to be concerned.

Chapter 4

Resource Allocation Improvement Approach

4.1 Introduction

This chapter introduces the reader to the resource allocation improvement process by giving a general overview of the method used. Chapter 5 discusses the Belief Network portion of this method in greater detail. Chapter 6 will demonstrate and validate the prototype program with the queuing example by Carmicheal (1987), the example presented in chapter 3.

The resource allocation improvement process basically involves the use of the simulation shell Simphony to generate and simulate a simulation model, using Microsoft Bayes Network to create and evaluate a belief network given inputs from the simulation run, and using a Visual Basic interface to initiate the model simulations, evaluate the results from the belief network, and modify the resources within the simulation model (Figure 4.1). The Access database serves two functions: first, it is used to record actions taken during the resource allocation improvement process; second, it is used at the end of the resource allocation improvement process to determine the configuration of the model that would yield either the preferred cost or duration.

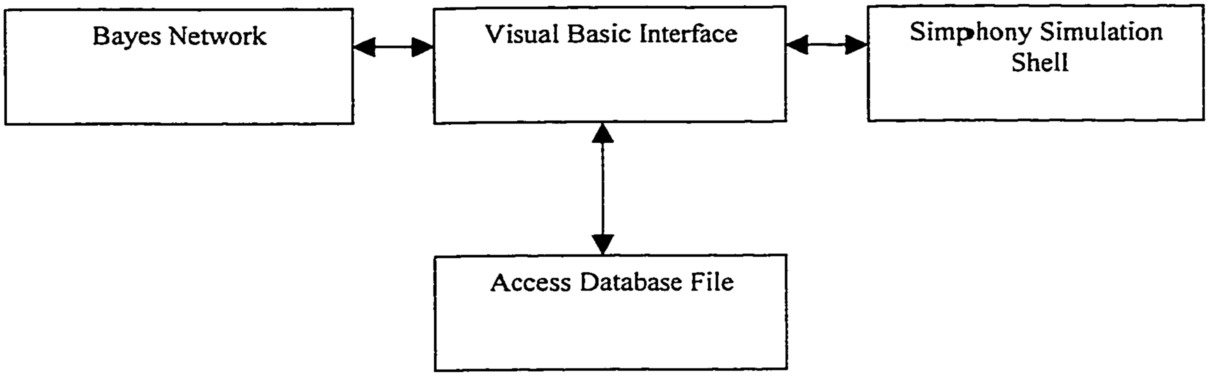


Figure 4.1 Resource Allocation Improvement Process Overview

The state of the simulation model is communicated to the belief network through performance indices that describe the queuing characteristics and resource utilization in the model. The belief network takes these indices, evaluates the queue situation in the simulation model, then identifies various problems in the queuing scenario, which are linked to recommendations that either increase or decrease the number of resources within a queue element in the model. The Visual Basic interface is used to select and carry out the appropriate recommendation for the iteration run, and also controls the resource allocation improvement process.

This chapter is broken down in the following manner: section 4.2 reviews work previously done in this area; section 4.3 gives an overview of how the Simulation Model, Visual Basic interface, Belief Network, and Access database files are used in the resource allocation improvement process; section 4.4 details how information is drawn from the simulation model and used by the Visual Basic interface; section 4.5 describes how the performance indices are evaluated so that they can be inputted into the belief network;

section 4.6 describes how the simulation model is modified; and section 4.7 concludes the chapter.

4.2 State of the Art

There have been several rule-based expert systems developed with the objective of resource management. These rule-based systems capture an expert's knowledge in a knowledge base. The knowledge base is structured such that navigation through the knowledge base by the inference engine is done via a series of if/then statements, and questions are posed by both the user and the system. McGartland and Hendrickson (1984) provided an example of how an expert system could be structured in order to be helpful in purchasing and inventory control. They suggested that an expert system of this type would prove useful in ensuring a project manager maintains appropriate inventory levels for a project. Sathi, Morton, and Roth (1986) developed CALLISTO, an expert system designed for aiding in the management of a project, focusing on project scheduling, control, and configuration problems. Using data detailing the project knowledge, activity durations, precedence, resources, and project constraints, the outputs from this system were suggested approaches to resource management, activity management, and configuration management. The resource management module outlined the specification and allocation of resources to carry out activities, including resources such as personnel, tools, parts, and so on. Hendrickson, Martinelli, and Rehak (1987) developed an expert system for masonry construction duration estimates called

MASON. This system estimated the duration of the activities, given a specified crew composition. The system would also suggest improvements to the crew composition so as to achieve better productivity. These recommendations were based on heuristic rules encoded within the knowledge base of the system. Hendrickson et al. (1987) developed Construction Planex, which was an expert system that generated project networks, cost estimates, and schedules. In so doing, Construction Planex suggested construction crew and equipment selection in order to best carry out activities in the project. Equipment and crew selection was done based on site conditions and the heuristic rules encoded in the knowledge base of the system. This system also allowed for a module that would make recommendations for crew composition to improve task productivity, as in MASON. Alkass and Harris (1988) developed ESEMPS (Expert System for Earth Moving Plant Selection) an expert system for the selection of equipment for road construction projects. In addition to equipment selection, this system also determined the appropriate number of pieces of equipment for a project for its given site conditions, based on conventional calculation methods. Amirkhan and Baker (1992) went one step further and developed an expert system for the selection of equipment for earthmoving operations. This system was a step ahead of Alkass and Harris (1988) as it was not limited in its domain to only road construction projects. It also gave an indication of the number of pieces of equipment required based on conventional calculation methods and the amount of earth to be hauled. The last rule-based expert system worth mentioning is the conceptual system for underground subway station rehabilitation projects presented in El-Choum and Rumala (1997). This system was designed to allocate tasks needed in an underground subway rehabilitation project and to select the appropriate amount of material, equipment, and crew/operators for the specific tasks within the project. The selection of material, equipment, and crews were based on historical data gathered from various project managers and experts, which was captured in the knowledge base using if/then statements.

Varis (1998) used a belief network approach for optimization and parameter estimation for resource and environmental management. His approach involved two distinct layers: the first layer was called a state layer and was composed of equations, or their equivalent, which modeled a desired system; the second layer was a probabilistic layer composed of a belief network that described external data not described by the model. This external data could be knowledge, experience, data, stated performance goals, etc. All this external data had a determined prior probability; for a specific state of the external data, there was a belief or probability that the state of the data was true given no other evidence. When evidence was contributed so as to add or remove confidence in this belief, the new belief in the state of the data was said to be a posterior probability. This is explained in further detail in the next chapter. The validation of the state layer was dependent on the posterior probabilities of the belief network being equal to the prior probabilities. Thus, the differences in the two probabilities in the probabilistic layer measured error in the state layer. The end result of this system was an iterative optimization process that involved the two layers communicating with each other until the modeling of the system had satisfied both layers.

The work that most closely follows this is that done by McCabe (1997), where an automated approach for construction performance improvement using simulation and belief networks was designed. This system was composed of two modules, a simulation module and a belief network module. Integration of the modules was provided via Visual Basic programming, and user inputs for the system consisted of a simulation model and the project constraints. Running the system commenced the iterative process where the simulation module would run a simulation of the model and pass performance indices to the belief network module, which would then identify and recommend modeling improvements to the simulation module. In this system, the belief network was fixed such that it was part of the system and its recommendations were not seen as input. Thus the belief network portion of the optimization process was rigid in construction and could not be altered with varying modeling scenarios.

McCabe (1997) also privileged the use of belief networks in the construction performance improvement process over neural networks, genetic algorithms, rule-based expert systems, and fuzzy logic. The rule-based expert systems and neural network input-output structure were found to be too structured and therefore unsuited to the changing states of the input variables. Belief networks proved flexible by having the ability to accept evidence at any time in the system and by being able to carry out an evaluation even with the absence of certain inputs. Creation of belief networks is done using expert opinion rather than historical data, another advantage belief networks have over neural networks. Genetic algorithms require no data for their development, although significant resources would be required to develop the generic objective functions.

Lastly, McCabe identified the ease in which a belief network can be constructed or modified as an additional reason for their superior suitability for the construction performance improvement process. The graphical nature of the belief network allows additions to the network to occur with little impact on the existing network, while changes to a Neural Network require a complete retraining of the network and changes to a rule-based expert system require meticulous evaluation of the rule base.

4.3 Overview of Approach and General Algorithm

The resource allocation improvement process views the activities in the simulation models as interactions between servers and customers. Therefore, the resource allocation improvement process improves the simulation model by observing the interactions in the model and then manipulating the number of customers or servers. For this prototype, in an effort to keep the system generic, the only manner in which the servers and customers can be manipulated is by increasing or decreasing their number. If one were to become more domain specific, such as building a resource allocation improvement process for a truck and shovel simulation template, then the servers and customers could take on a size aspect as well. That is, the process could change the size of the loader from 25 cubic meters to 35 cubic meters in addition to changing the number of loaders.

The resource allocation improvement process can be described as an iterative process that cycles between simulation, getting recommendations from the belief network, evaluating the recommendations, and changing the model (Figure 4.2).

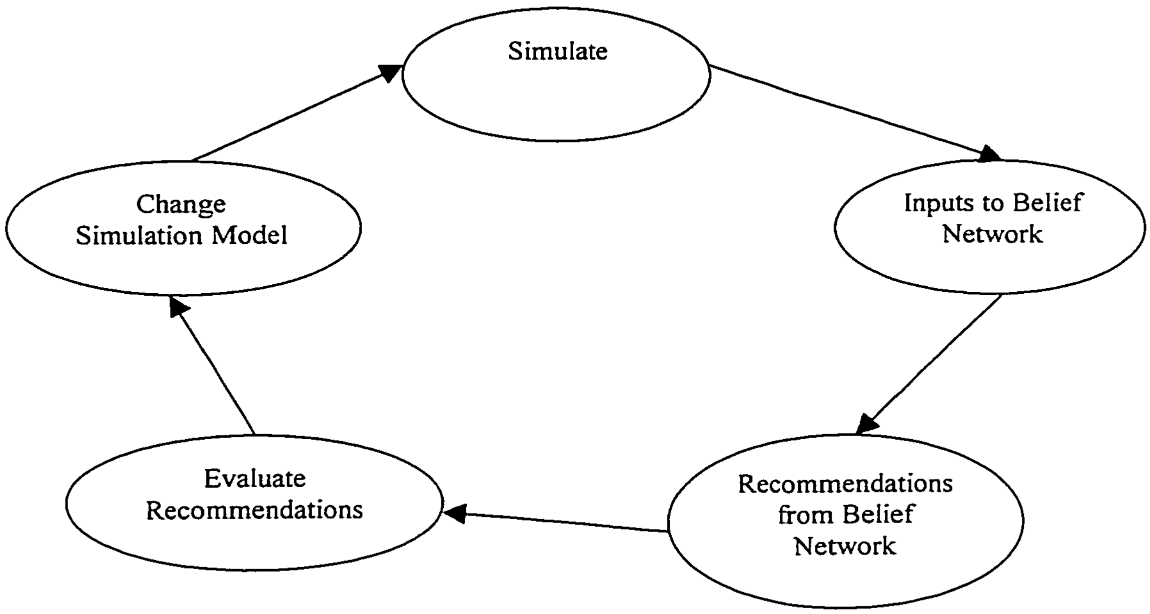


Figure 4.2 Iterative Resource Allocation Improvement Process

The user builds both the simulation model and the belief network. The queue nodes intrinsically generate the output performance indices for the simulation model, however, this causes the belief network to be slightly more complicated in its creation. The created belief network must have nodes labeled in a certain manner so as to be identified as input nodes or output nodes. The nodes named "CD", "QW", "QL", "SU", "Project Cost" and "Duration" are used as input for the Customer Delay Index, Average Queue Wait, Average Queue Length, Server Utilization, the project cost allocation value, and the project duration allocation value respectively. The "Project Cost" and "Duration" nodes are used to indicate to the belief network whether to improve resource allocation with respect to cost or duration. The output nodes must have their category name labeled as "Output" so that they can be recognized (Figure 4.3). A sample network is presented in Figure 4.4.

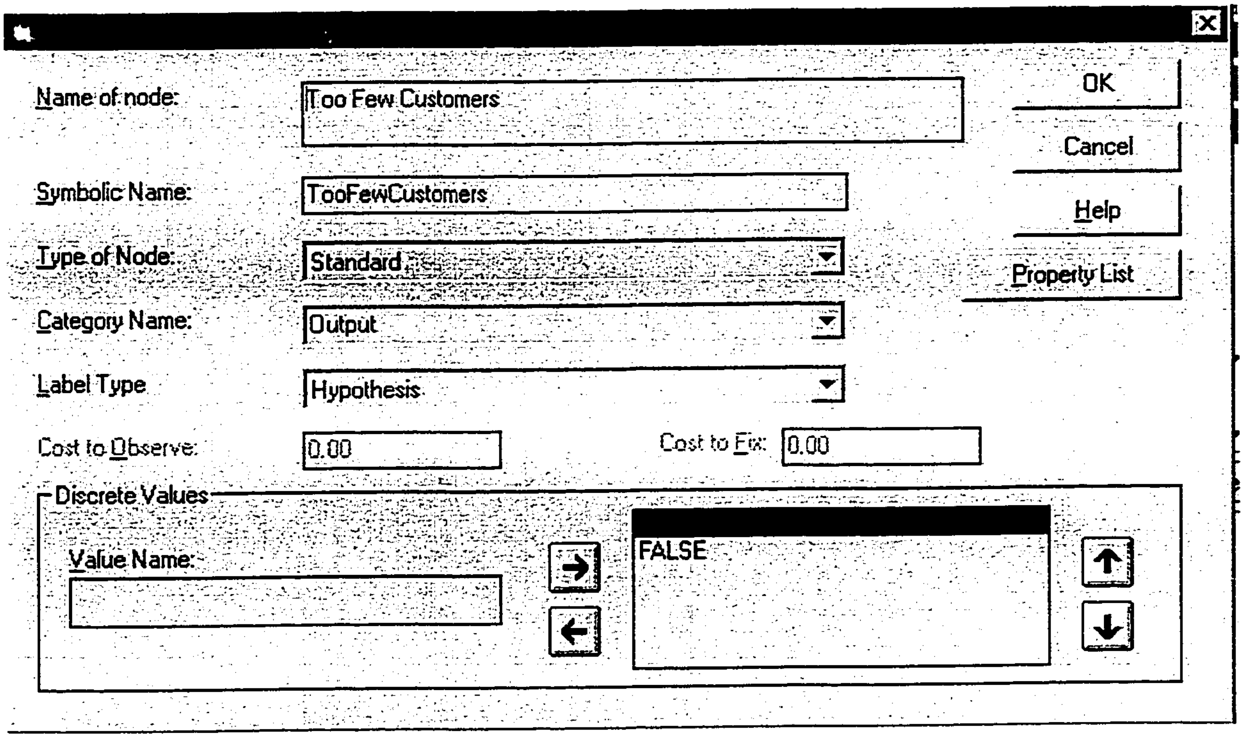


Figure 4.3 Properties Box For Belief Network Node

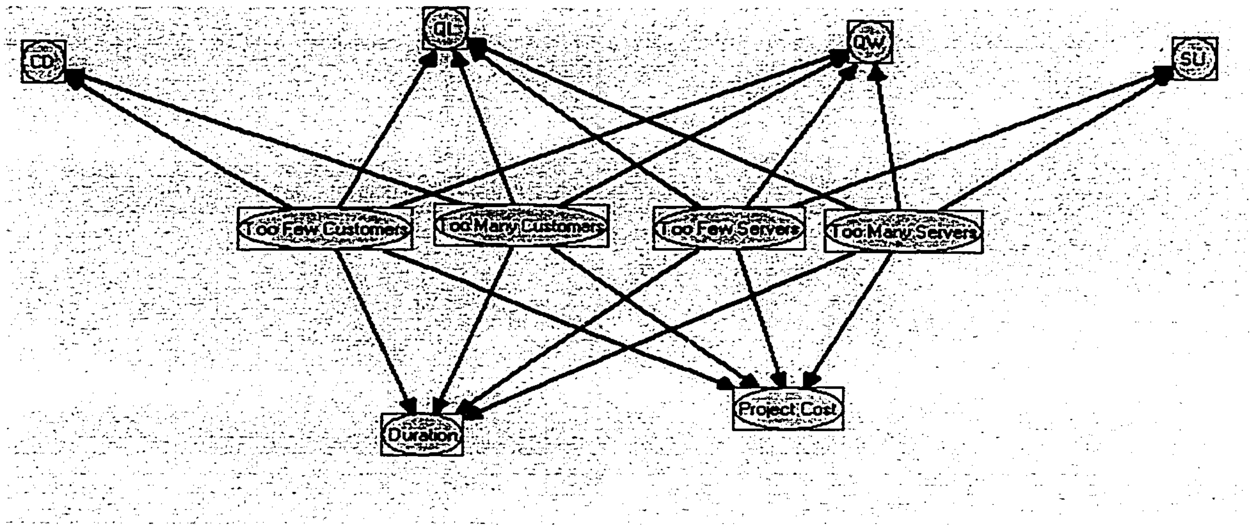


Figure 4.4 Belief Network

The output nodes in the belief networks represent possible problems that can be identified given the inputs supplied to the network. Exactly how the network works is explained in the next chapter. For now it is important to understand that the outputs from the belief network are the identified problems for each queuing situation, given the supplied information. These problems in the belief network are linked to suggested recommendations to improve the simulation model. At the start of the resource allocation improvement process, the user links each of the problems to one of the possible recommendations. Although the user is allowed to create the network, thereby creating any problem node possible, the recommendations are hard coded. Again, this was done to keep the system generic. For this prototype, there are only four possible recommendations to which these problem nodes can be linked: increase a server, decrease a server, increase a customer, or decrease a customer.

After each simulation run has been completed, each queue element within the model calculates its performance indices. These indices, along with their tolerances and the information used to identify the queue in which the information came from, are then recorded for each queue element in the model. Thus every queuing situation occurring in the simulation can be described and evaluated by the belief network. After obtaining this information, the Visual Basic interface evaluates the queue elements' performance indices against their tolerance values such that the state of the input nodes in the Belief network can be determined. With the exception of the Customer Delay Index, the performance indices can have three states: below the desired tolerance, acceptable, and above the desired tolerance. The possible states of the Customer Delay Index are either acceptable or above the desired tolerance. The next step is to set the input nodes in the belief networks to their appropriate states depending on the values of the performance indices. The Project Cost and Duration nodes are set to either yes or no depending on which the model is being taken into consideration for. These nodes act like a switch; in other words, if cost is being taken into consideration, the Project Cost node is set to "Yes" and the Duration node is set to "No". Both of these nodes cannot be set to "Yes" at the same time.

Thus, for each queue element in the model, its performance indices are inputted into the belief network and recommendations are made for the queue element based on its identified problems. Each of these problems is linked to one of the four possible recommendations. The next step in the process is to shorten this larger list of recommendations to a more viable list of recommendations from which to choose. The recommendations can be eliminated in one of three ways: due to a low belief value, contradiction, or inability to be carried out.

The first of the recommendations to be eliminated are those that do not have a belief value over the user-specified value. The belief network identifies each problem with a belief value, which is the probability of the identified problem being true. The user is allowed to specify the cut off belief value at which the identified problems can be considered for recommendations. Thus, if a recommendation is found for a problem with a belief of 40% and the user defined cut off belief value is 50%, that recommendation is eliminated from further consideration (Figure 4.5).

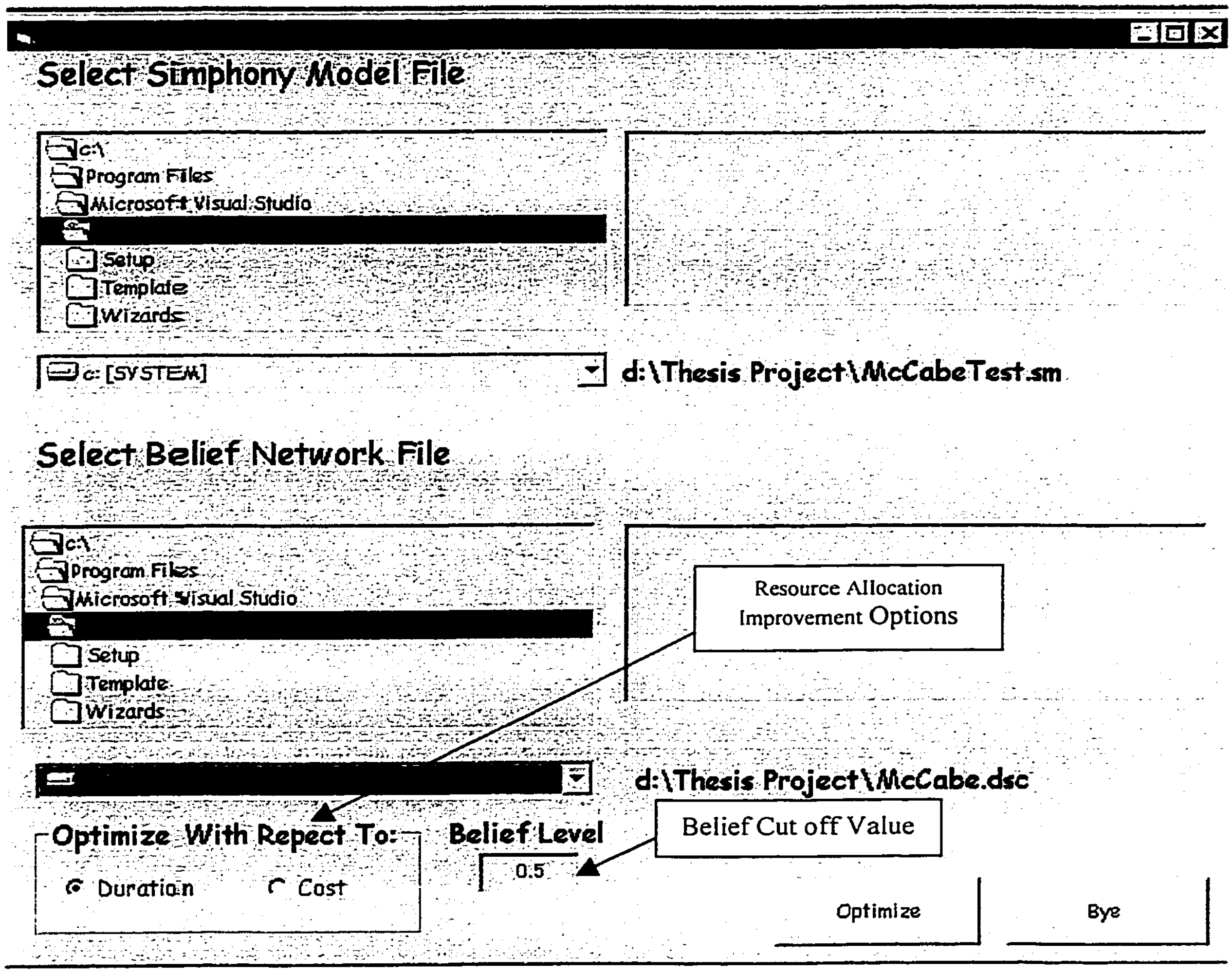


Figure 4.5 Visual Basic Interface Start Up Form

The next check performed looks to see if any of the remaining recommendations contradict each other by performing a search to see if contradicting recommendations are occurring. For example, one recommendation for Server Queue Element #10 could be to increase its customers, while another recommendation for Server Queue Element #10 could be to decrease its customers. These recommendations contradict each other, recommending opposite actions for the same queue element. The solution to this problem is to disregard both these recommendations. Thus, any contradicting recommendations are omitted from the recommendation list.

Lastly, any recommendation that suggests an action that cannot be allowed is eliminated from the recommendation list. As an example, Server Queue Element # 20 could have a recommendation to increase its resource number from 2 to 3. When creating the simulation model, the user may have specified that Server Queue Element # 20 could only have a maximum of 2 resources. Therefore this action cannot be carried through and is eliminated.

Once the finalized list of recommendations is attained, the recommendations are sorted from highest to lowest according to their belief value. At the end of the first resource allocation improvement iteration, the user is presented with the list (Figure 4.6). The program will have automatically chosen the recommendation with the highest belief and will ask the user to confirm this choice. The user has the ability to override this choice and choose another recommendation, to confirm every iteration, or to let the system automate the process. If they let the computer automate the process, the recommendation with the highest belief value will be chosen for each iteration. This automated process can be terminated in one of three ways. The first is by achieving an improved model where the system can make no further recommendations. The second manner is by reaching the maximum number of iterations specified by the user when the automated process is selected. The last situation where the process will terminate is when the effectiveness of the simulation has reached a plateau. When specifying the resource allocation improvement process the user has the option to specify a tolerance value that the change in cost or duration, depending on what is being taken into consideration, *must exceed* at least *once* in the last five iterations in order for the process to continue (Figure 4.5). If this tolerance value is not exceeded, the process will terminate.

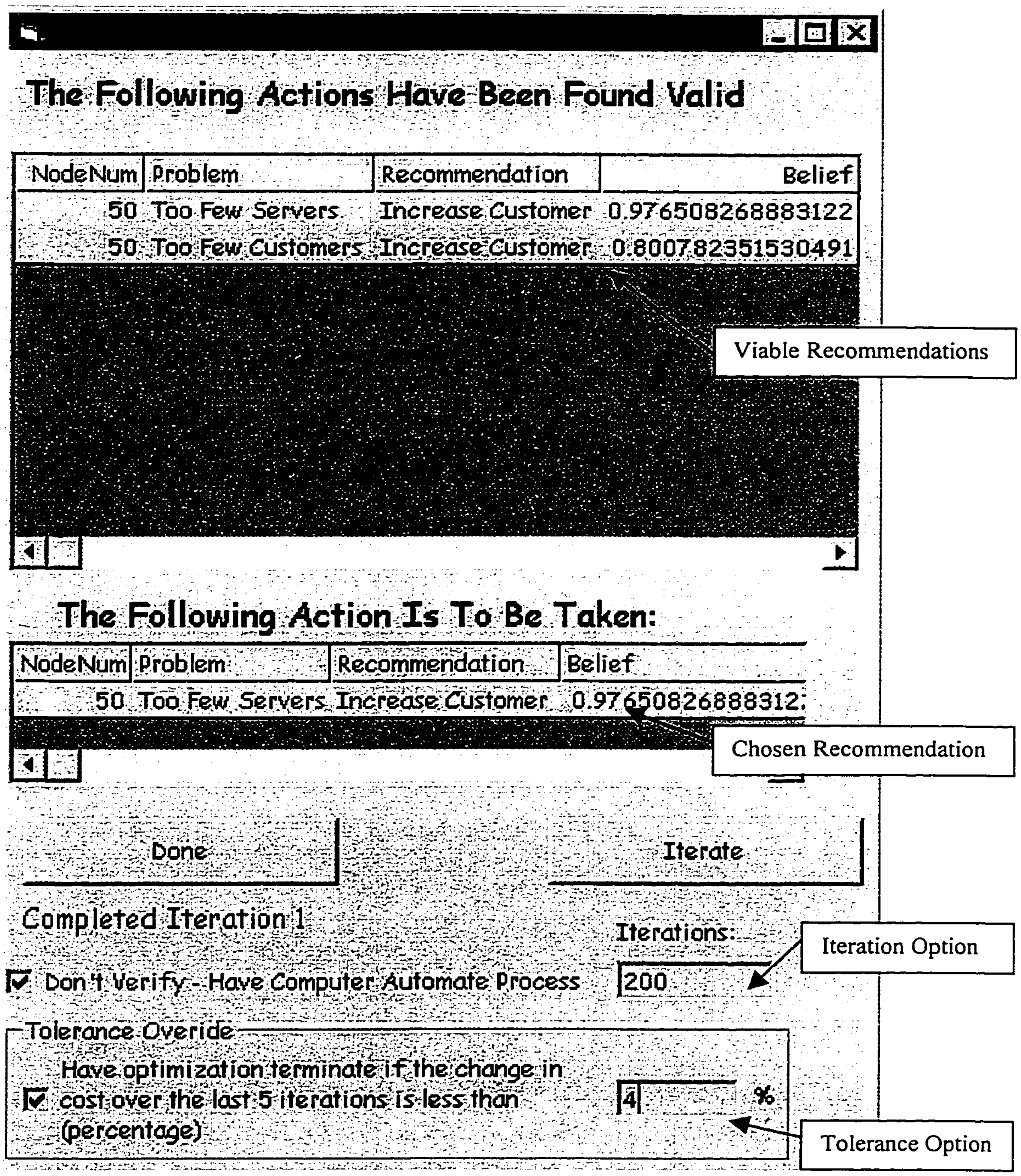


Figure 4.6 Iteration Results Form

4.4 Reading The Simulation Model

The elements of concern for the resource allocation improvement process are the Server and Customer Queue Elements, as these elements contain all of the information required for the resource allocation improvement process. They contain the performance indices that describe the state of the simulation model to the belief network. They contain the loop numbers for the customers and servers that may need to be increased or decreased depending on the recommendation of the belief network. They also contain information regarding whether or not an element is permitted to be increased or decreased.

When the resource allocation improvement process is compiling its initial list for recommendations, it will find every queue element in the simulation model, collect information from the queue, and get recommendations for the model based on information from that queue. The information differs between the two types of queues, varying mainly with regard to the collection of the performance factors. Server Queue Elements use the server utilization statistic while the Customer Queue Elements use the Customer Delay Index. The information collected from the queues that is common to both the Server Queue and the Customer Queue are the queue element type, the element identification number, the Average Queue Wait and its tolerances, the Average Queue Length and its tolerances, the resource minimum value, the resource maximum value, whether the resource number can be increased, whether the resource number can be decreased, the loop numbers of the attached Combi Elements, and the Queue Element loop numbers.

4.5 Evaluating the Performance Indices

As mentioned before, there are four indices used to describe the model to the belief network: the Average Queue Wait, Average Queue Length, Server Utilization, and Customer Delay Index. They are calculated as follows:

The Average Queue Wait is the average wait experienced by all entities as they come to a specific queue. It is calculated by dividing the sum of all the waits experienced by all the entities that came to that queue during the simulation by the number of entities that came to that queue.

$$AQW = \frac{\Sigma \text{ Wait Experienced By Entity in Queue}}{\text{Number of Entities to Come to Queue}} \qquad (4.1)$$

The Average Queue Length is the average length of entities waiting at a queue for a given simulation run. It is calculated by dividing the sum of the various queue lengths, weighted by the time they were at that given length, by the simulation time.

$$AQL = \frac{\Sigma \text{ Queue Length * Time At That Length}}{\text{Simulation Time}} \qquad (4.2)$$

The Server Utilization is the ratio of the server's productive time compared to the total time the server was required for the project. It is calculated by dividing the sum of the server's productive time by the total time the server was required for the project.

$$SU = \frac{\Sigma \text{ Server Productive Time}}{(\text{Server Finish Time - Server Start Time})} \qquad (4.3)$$

The Customer Delay Index is the ratio of the average delay time experienced by an entity during its cycle to its entire cycle time. It is calculated by dividing the sum of the customer's delay or waiting time by its cycle time.

$$CDI = \frac{\Sigma \text{ Customer Experienced Delays For Cycle}}{\text{Cycle Time}} \qquad (4.4)$$

These indices are generated by the simulation, based on the user-specified tolerances that set their acceptable limits. With the exception of the Customer Delay Index, which only has an upper limit, each of these indices have an upper and lower tolerance. The user is asked to specify the 90^{th} percentile for the upper limit and the 10^{th} percentile for the lower limit, which are a safety margin to ensure that the tolerances specified by the user in the simulation are not exceeded in reality. This margin of safety is in place so as to safeguard against deviations occurring in reality due to the inherit randomness associated with construction. This margin of safety simply ensures that the results from the simulation and the resource allocation algorithm will not exceed what occurs during actual construction.

4.6 Modifying The Model

There are four possible ways in which a model can be modified by increasing or decreasing a customer or server: a Customer Queue Element requesting an increase/decrease in the number of customers, a Customer Queue Element requesting an increase/decrease in the number of servers, a Server Queue Element requesting an increase/decrease in the number of customers, and a Server Queue Element requesting an increase/decrease in the number of servers. Each recommendation is carried out in a different manner and is explained in detail in the remainder of this section.

If a Customer Queue Element requests an increase or decrease to the customer number, the resource itself will be increased or decreased by one as required. No other Customer Queue Elements will be affected.

If a Customer Queue Element requests an increase or decrease to the server number, all of the allowable servers in the loops connected to the Customer Queue Element will be increased or decreased by one. The allowable servers in these loops are those that can be increased or decreased without compromising any user-defined resource limits. Each Customer Queue Element can be attached to four loops, therefore this action can only affect servers in up to four loops.

If a Server Queue Element requests an increase or decrease to the customer number, all of the allowable customers that are connected to the succeeding Combi Element loops will be increased or decreased by one. This action is taken since if a server needs an increase in the number of customers for an interaction at a Combi Element activity, it requires an increase in the number of all incoming customers, given that it is possible to do so. Each Server Element Queue can be connected to four loops, which implies that each Server Queue Element can be connected to four Combi Elements. Each Combi Element can be connected to four other loops with one of these loops being connected to the Server Queue Element. Thus, in the most extreme case, a Server Queue Element could increase or decrease the customers in 12 loops.

Lastly, if a Server Queue Element requests an increase or decrease to the server number, the resource itself will be increased or decreased by one as required. No other Server or Customer Queue Elements will be affected.

At the end of the improvement process, once the process has been terminated by one of the three possible situations, a search of the information in the database occurs. Depending on what the model is being taken into consideration, the computer will search the database for the iteration that resulted in the lowest cost or duration. Once this iteration has been found, the computer will find the resource allocations to the queues for that iteration and set the simulation model back to those settings.

4.7 Conclusion

A general overview has been given for the resource allocation improvement process. This method involves using performance indices to describe the state of the simulation model to a belief network, which is used as an expert system to make recommendations to the simulation model. The simulation model is thus viewed as a group of interactions between server entities and customer entities and their performance factors are used to describe their utilization, delays, and the manner in which they queue during the simulation.

Significant detail was given with regard to the resource allocation improvement algorithm. This iterative process consisted of simulating a model, calculating the required performance indices, using these indices to get recommendations from the belief network, sorting through these recommendations to get a viable list of recommendations, and finally implementing the recommendation.

Lastly, it was mentioned how the preferred resource allocation was determined. Upon completion of the process, the system finds the iteration with the lowest cost or duration and sets the simulation model back to the resource allocations for that iteration. The database file allows the user to view all of the information regarding the resource allocation improvement process.

In regards to further development for the resource allocation improvement process, additional development could be done to address the selection of the recommendations for the model improvement. Firstly, a different method of selection could be devised in which the recommendation with the highest belief is selected regardless of any conflicts this recommendation would have with any other recommendations. A second recommendation would be to develop a method in which combinations of recommendations could be investigated and applied to the model for a single iteration.

Chapter 5

Belief Networks

5.1 Introduction

The decision-making tool used in the resource allocation improvement process is heavily dependent on a belief network. Minkarah and Ahmad (1989) defined an expert system as computer program having expertise, symbolic reasoning, depth, and knowledge. In this system, the belief network displays all these characteristics. It takes in information, evaluates it based on an expert's development of the network, and reasons to identify problems occurring in the simulation model based on the inputs. The method of developing a belief network described here has been done in such a way that the network is flexible enough to allow the user to build creativity into the belief network's design so that any belief network can be developed to perform as its user desires. This chapter is an introduction to belief networks and how they are created and used in the resource allocation improvement process.

This chapter is broken down in the following manner: section 5.2 gives a review of past achievements in this field of study; section 5.3 introduces the reader to belief networks, their function, and the theory behind them; section 5.4 explains how to construct a belief network for diagnostic purposes; section 5.5 explains how corrective actions to the simulation model are determined from the belief network; and section 5.6 sums up the information presented.

5.2 State of the Art

According to McCabe (1997) Bayesian or Belief Networks were first developed at Stanford University in the 1970's. They fell out of popular research in the 1980's and have experienced a resurgence in the 1990's. Other than McCabe's efforts in this field of research, no other endeavors using Belief Networks for simulation resource allocation improvement have been found in this or any other field of research. Though no such endeavors exist, there have been some related efforts using Bayesian statistics in civil engineering. Kriviak and Scanlon (1986) used a Bayesian statistical approach to combine small or large amounts of data to determine the compressive strength of concrete. Sorensen and Engelund (1997) used a Bayesian statistical approach to parameter estimation for modeling chloride ingress into concrete structures such that a maintenance and repair strategy could be developed. Der Kiureghian (1990) used a Bayesian statistical approach to assess uncertainty in the mathematical modeling of the behavior of a structure and the effects of the uncertainty in structural reliability. Applications of belief networks can be found in other areas of research such as environmental engineering (Chong and Walley 1996) and medicine (Charniak 1991).

5.3 Belief Network

A belief network is a series of variables or nodes linked together in a network by arcs (Figure 5.1). The nodes represent the variables that describe a domain of interest while the arcs represent the relationships between the nodes within the networks. Nodes that are linked together have a direct relationship with each other. The condition of each node is described by its state, and the states can vary according to the variable being described. For example, a node called "Hot" could have the states of "Yes" and "No". A node called "Temperature" could have the states of "Cold", "Warm", and "Hot". A change in state of a variable will impact only the variables directly connected to it. Nodes that are not directly connected are independent of each other; this is also known as d-separation. Changing the state of a node will have no direct effect on a node independent from it.

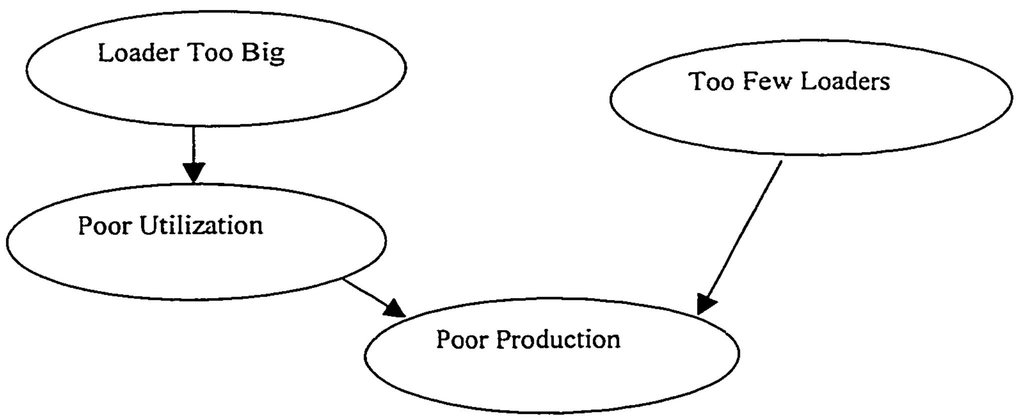


Figure 5.1 Belief Network

The function of a belief network is to determine the belief or probability of a proposition given various bits of information regarding the states of other variables in the network.

For example, if there are four nodes in a network labeled A through D that can either be true or false, one may want to know the belief of Node A being true, given that Node B and Node C are true and Node D is unobserved. The resulting belief in Node A, given the information of the other nodes, is a probability of A being true, and should not be considered a degree of Node A's truth. For example, given the information, its has been found that Node A has a probability of 80% of being true, but this belief in Node A does not mean that Node A is 80% true.

The resource allocation improvement method uses the belief network expert system to facilitate decision-making. The user creates the network and thereby controls how the network functions. The inputs for the networks are the performance factors used to describe the queuing situations occurring in the simulation run, and the outputs from the network are identified problems in the simulation model. These problems are then linked, in the Visual Basic interface, to recommendations for model improvement. For this prototype, an attempt has been made to keep this system domain generic, therefore these recommendations only manipulate the number of resources in the system. More specialized systems could go further and manipulate the capacity and types of resources in the system.

5.3.1 Definition

Russell and Norvig (1994) defined a Belief network as a graph in which the following hold:

1. A set of random variables make up the nodes of the network.
2. A set of distinct links or arcs connect pairs of nodes. These arcs have a direction; an arrow going from node A to node B indicates that node A has a direct influence on node B.
3. Each node has a conditional probability table that quantifies the effects the parent nodes have on it.
4. The graph is a directed acyclic graph-(DAG). Thus, there are no cycles to be found in the graph.

Belief networks can therefore be described as a set of nodes and arcs, where the nodes are variables that describe the domain and the arcs represent the relationships between these variables. These arcs are directional and indicate which parent nodes, situated at the beginning of the arrow, have a direct effect on their children nodes, situated at the end of the arrow. The network itself is acyclic, meaning that no cyclic loops can occur in the network.

A belief network can best be described as a snapshot of the joint probability distribution. The joint probability distribution, also known as the joint, specifies the probability assignments to all propositions in the domain. The joint specifies the probability for every combination or query of the various states of the variable in the domain, while the belief network specifies just one probability given the current state of the variables in the domain. The advantage of using a belief network over the joint is in the number of calculations required to define it. Given that the variables in a domain have two possible states, either true or false, the number of calculations required to define the joint would be 2^n, where n is the number of variables in the domain. Thus as the number of variables in the domain grow linearly, the number of equations required to define the joint grow exponentially. To define the belief network, one must only define the conditional probability tables for the relations between the nodes.

The belief network operates using conditional probabilities and Bayesian statistics to determine the probability of a proposition being true or false. This probability is a measure of one's belief in a proposition and should not looked on as a measure of truth. Probability is simply a method of summing up the uncertainty one may have when reasoning as to the likelihood of an event occurring. Thus a belief of 80 % in a proposition being true would indicate an 80 % confidence in that proposition being true and a 20 % degree of uncertainty.

5.3.2 Conditional Probability and Bayes's Theorem

There are essentially two basic type of probabilities: prior and posterior. Probabilities determined before any evidence or information is given are considered prior probabilities.

A probability that is determined after evidence has given insight to the proposition is known as a posterior or conditional probability. The difference between the two probabilities is best exemplified in an example using a deck of cards. The problem at hand is determining the probability of a card drawn from the deck being the Ace of Spades. The prior probability for the problem is 1 card out of 52 or 2%. Now say it is known that the card drawn is a spade. The conditional probability of the card being the Ace of Spades, since it is known that the card is a spade, is 1 card out of 13 or 8%. Thus the difference between a conditional probability and a prior probability is that the conditional probability is dependent on the values of other variables, while a prior probability is not.

Bayesian statistics deal with analyzing conditional probabilities. Belief networks make use of Bayes's Theorem for their calculation. The following is an expanded form of Bayes's theorem that is able to handle multiple influences.

$$P(B_j|A) = \frac{P(A|B_j) * P(B_j)}{\sum_{k=1}^{n} P(A|B_k) * P(B_k)} \qquad (5.0)$$

The notation $P(B_i|A)$ used for this equation is the probability of B being true given that A is true. If a variable were to be false, its notation would be $\neg A$.

5.3.3 Evaluation of a Belief Network

Belief networks can be classified into singly-connected networks and multiply-connected networks (Figure 5.2 and Figure 5.3). The difference between these two categories is that there is only one path between any two nodes in a singly connected network while a multiply connected network can have nodes with two or more paths connecting them. For example, in Figure 5.3 one can see that there are two paths from node A to node D. Singly-connected networks allow for an exact solution while multiply-connected networks have approximation methods which convert the network to a singly-connected network that can then be solved.

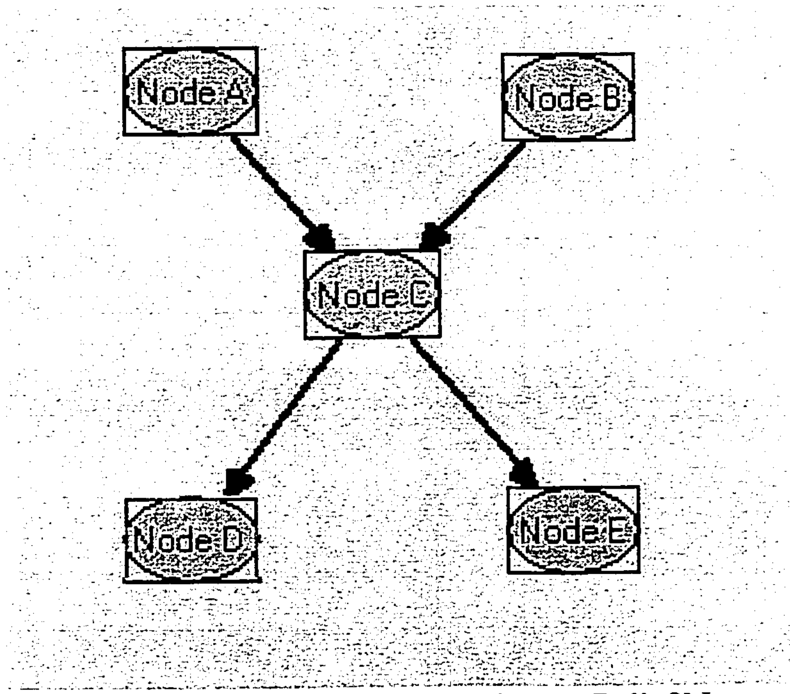


Figure 5.2 Singly-Connected or PolyTree Belief Network

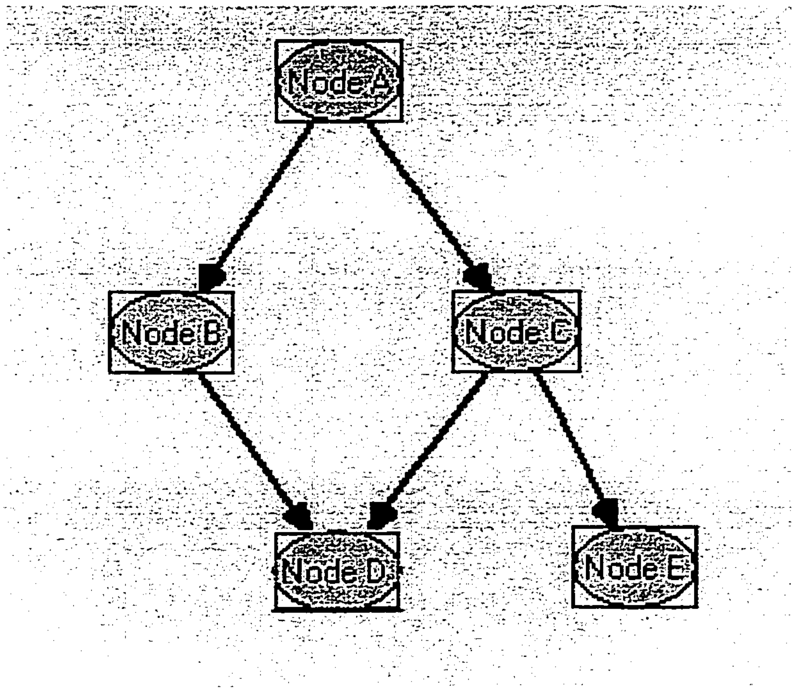


Figure 5.3 Multiply-Connected Belief Network

For the most part, the calculation of a belief network is complex and requires many calculations. An example of an evaluation of a simple network can be found in McCabe (1997). Software programs, such as Microsoft Bayes Network, aid in solving these networks quickly and easily. The fact that it is Visual Basic integratable makes it very applicable to the problem at hand.

5.4 Creating a Belief Network for Diagnostic Purposes

Poole et al. (1998) and Russell and Norvig (1994) outlined a method for constructing a belief network. It is as follows:

1. Define all relevant variables that describe the domain.
2. Choose an ordering for the variables. Root variables, which are those that have no parents, are ordered first, followed by those variables that would have the root variables as parents. Nodes that would have the previously listed nodes as parents are listed next and so on, until all of the variables are listed.
3. As each variable is removed from the list and placed in the network, arcs should be draw from previously placed nodes to the recently placed node. Once this is done, the conditional probability table for the node should be defined.

In addition to the above method, there are two other constraints on how a belief network is to be constructed for this system that deal with the inputting and outputting of information to and from the belief network. The first constraint is that each network is to be constructed with six specific input nodes. These nodes are to be named "CD", "SU", "QW", "QL", "Project Cost", and "Duration". The second constraint is for the output nodes. The user is allowed to define the names and numbers of the output nodes, but the nodes must have their "Category Name" labeled as "Output" in order to be identified.

Each output variable is linked to an action for simulation model recommendation. When the resource allocation improvement process is run from the Visual Basic Interface, the user is presented with the name of each output node and asked to pick a recommended action for that node at the start of the first iteration (Figure 5.4). This will link the problems to the recommended actions for the resource allocation improvement process.

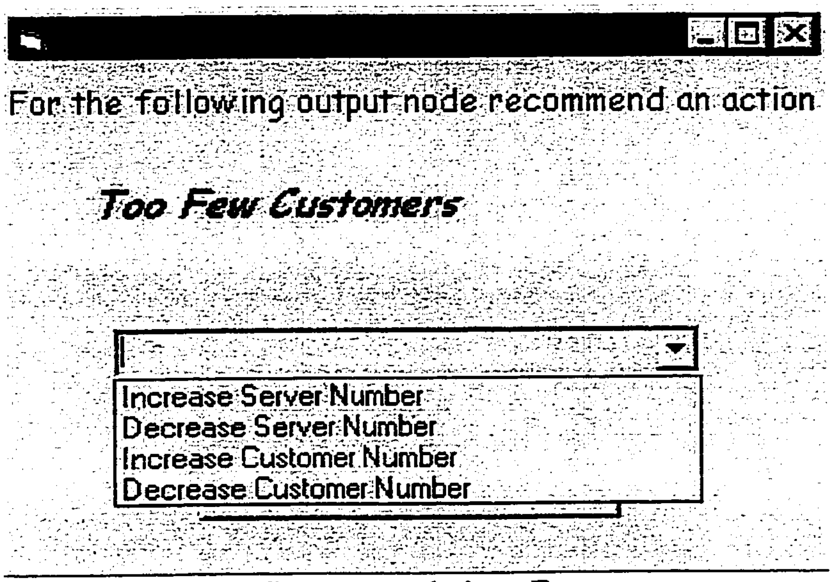


Figure 5.4 Output Recommendations Form

5.4.1 Performance Variables

The performance variables are the input variables for the network. For the prototype, these variables are the nodes that use the Customer Delay Index, Server Utilization, Average Queue Wait, Average Queue Length, Project Cost and Project Duration as inputs. These nodes must be labeled as "CD", "SU", "QW", "QL", "Project Cost", and "Duration" respectively. The various states for these elements are listed in Table 5.1. The second column of Table 5.1 lists the value used to set the state of the node from the Visual Basic interface, which are hard-coded in the system. The user should ensure that their belief network is set up so that the values for the states match those in the Visual Basic interface. If a variable is unobserved, its state is set with the unobserved value of 22222.

CD	0	$CD<CD_U$
	1	$CD>CD_U$
SU	0	$SU_L<SU<SU_U$
	1	$SU<SU_L$
	2	$SU>SU_U$
QW	0	$QW_L<QW<QW_U$
	1	$QW<QW_L$
	2	$QW>QW_U$
QL	0	$QL_L<QL<QL_U$
	1	$QL<QL_L$
	2	$QL>QL_U$
Project Cost	0	Good
	1	Consider
Duration	0	Good
	1	Consider

Table 5.1 States for Performance Variables

The evaluation of the performance indices occurs in the Visual Basic interface. The evaluation involves the comparing of indices against their tolerances to determine what state value the node state must be set to. Once all of the states are set for the nodes, the belief values of the output nodes can be determined. One of the advantages of belief networks is that not all the input variables must be observed in order to evaluate the network. If a node is unobserved its belief is based on its prior probability. The only cases where a nodes belief can be based on conditional probabilities is where the node state is changed or where a node with a direct relationship to that node has its belief value being affected. For example, in the case of a Customer Queue Element, the state of the Server Utilization node can be set to unobserved using the state value of 22222.

5.4.2 Causal Variables

The causal variables are the output variables for the network. These nodes are recognized by the Visual Basic interface by their category name (Figure 5.5). All nodes with the category name labeled as "Output" will be identified as output nodes and will be linked to a recommendation by the Visual Basic interface. Any nodes that are not labeled as "Output" nodes will function as normal nodes within the belief network and will not have any interaction with the Visual Basic interface.

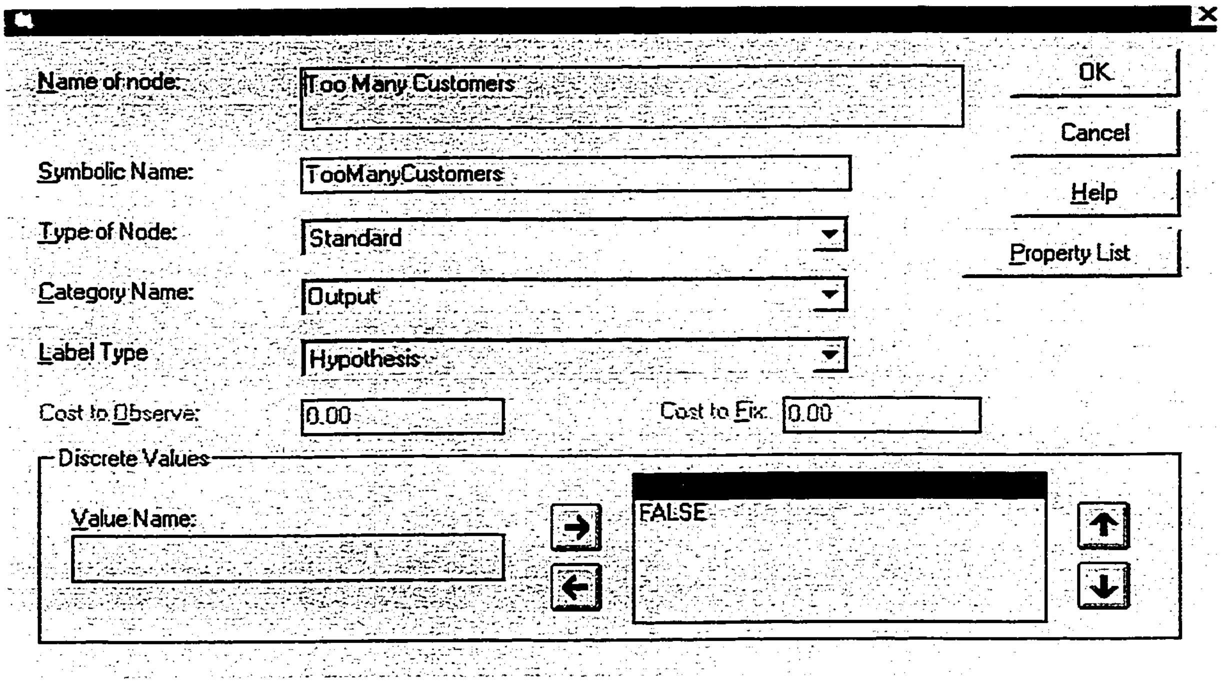


Figure 5.5 Properties Box For Belief Network Node

Upon the initial run of the resource allocation improvement process, all the nodes with the category name "Output" will be identified and the user will be asked to link each of these nodes to one of the four possible recommendations. The causal variables in the network are thus possible problems that can be identified given the various states of the performance variables. The states for the casual variable nodes are true and false.

5.4.3 Conditional Relationships

Conditional Relationships are the relationships between the nodes of the network. Only nodes that are directly related are linked. Nodes that are not directly linked together with a relationship are said to be independent of or d-separated from each other. Therefore the state of one variable that is not directly linked to another variable has no effect on it.

5.4.4 Conditional Probabilities

The conditional probabilities quantify the relationships between the parent nodes and their children and make up the conditional probability table for the child node. The conditional probability table is constructed by defining the probabilistic values for a child node for all possible combinations of the states of their parent nodes. An example of the conditional probabilities for a given network is given in Figure 5.6.

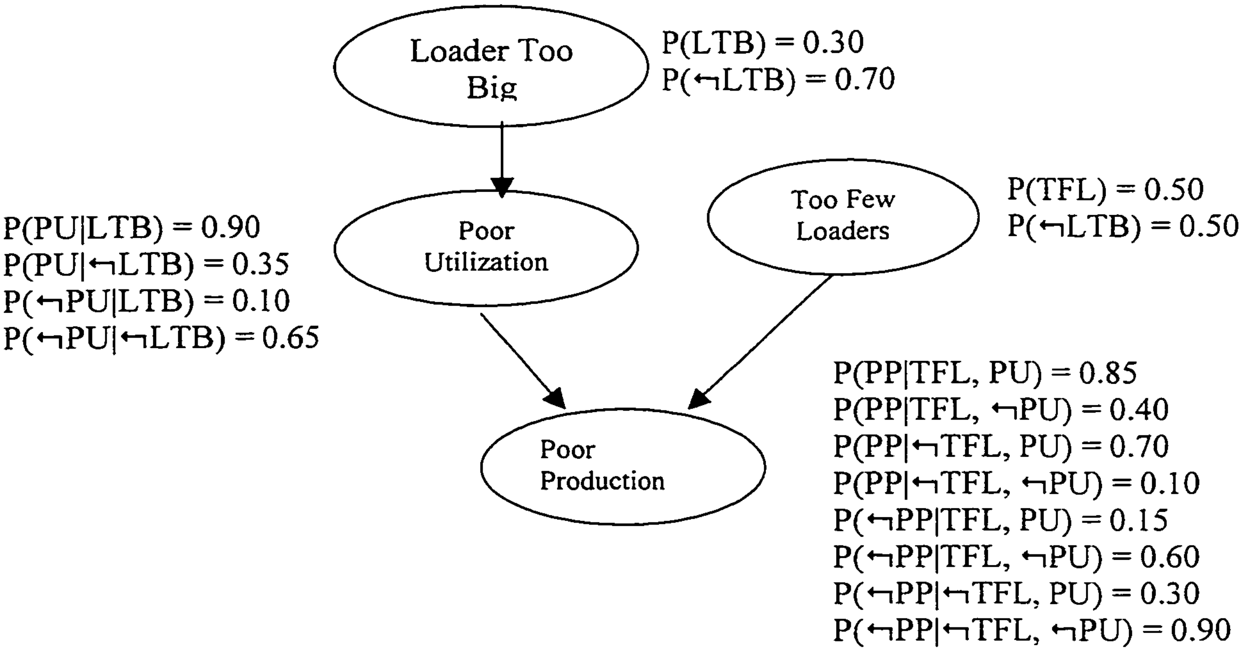


Figure 5.6 Belief Network with Conditional Probabilities

5.4.5 Validation of Belief Networks

The belief network that will be used for the validation example in chapter six is shown in Figure 5.7. This model is relatively simple with six input nodes and four output nodes, which identify the problems of "too many servers", "too few servers", "too many customers", and "too few customers". The cost of the server is to be double that of the customer and the belief network has been set to reflect this.

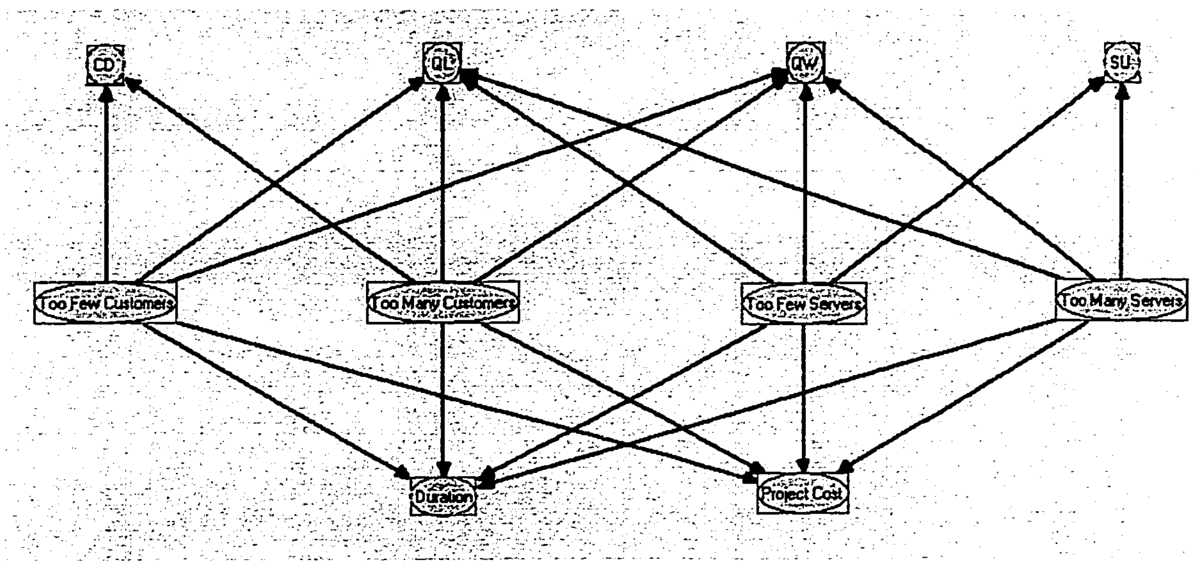


Figure 5.7 Belief Network Used for Resource Allocation Improvement Process

This belief network was evaluated and fine-tuned until satisfactory results were achieved. The input variables and their states have been set up as in Table 5.1. The conditional probability values are presented in Appendix A. The belief values for the various states of the input variables are found in Appendix B.

5.4.6 Determining Appropriate Corrective Actions

The belief network set up for this prototype is to be set up such that the recommendations are ranked hierarchically based on the belief for each one. Therefore the belief network's final output must be examined to ensure that these values are in fact hierarchical and not relative to each other in a single evaluation. In other words, if the simulation model is to be improved with respect to cost and the server is more expensive than the customers, the belief network should be set up such that the recommendations to reduce the number of servers is ranked higher than the recommendations to decrease the number of customers.

After the evaluation of all the queuing situations in the simulation model is complete, the recommendation list is complied. This list is then examined and revised such that only viable recommendations are left, as described in the previous chapter.

5.5 Conclusion

This chapter introduced the reader to belief networks and how they function for this research, which uses belief networks as an expert system. It takes in information, reasons with it based on network development by an expert, and identifies possible problems in resource allocation for a simulation model. The outputs from the belief network are the identified problems and the belief value associated with those problems. The Visual

Basic Interface takes these problems and their belief values and creates links between them to model recommendations for the improvement of resource allocation. The belief values associated with these recommendations are used to rank the recommendations from the strongest to weakest.

The belief network was described as a network consisting of nodes and arcs, where the nodes represent variables that describe a domain of interest and the arcs represent the relationships between those variables. A method of constructing a belief network to ensure that the network functioned as desired and remained a directed acyclic graph (DAG) was presented. In addition to the construction method, it was emphasized that the user must add the correct input nodes to the network and label the output nodes accordingly, such that the network is able to communicate with the Visual Basic interface.

There are several possible areas of improvement for this system. To allow greater flexibility in the modeling and use of belief networks as a resource allocation improvement tool, further work could be done to increase the flexibility and versatility of the entire resource allocation improvement method. One could to devise a manner in which the user could define and link their own performance indices and be able to create their own recommendations for the simulation model. This would allow other non-cyclic simulation programs, such as a gravel crushing operation and a dewatering simulation, to be easily applied to this method of resource allocation improvement.

Chapter 6

Integration of Simulation and Belief Networks

6.1 Introduction

This chapter introduces the reader to the Visual Basic interface that links all of the programs together and controls how they are used during the resource allocation improvement process. The user creates the belief network and simulation model outside of the interface before running the resource allocation improvement process. Once the model and the belief network are created, the interface asks for the input files, information regarding the files, and some parameters for the resource allocation improvement process. The rest of the process is left to the interface. Error handlers have been implemented and simplicity has been maintained to leave little room for user error or confusion.

A validation example is presented in this chapter for a simple queuing problem taken from Carmicheal (1987). The problem is the same one used to validate McCabe's (1997) Automated Modeling Approach for Construction Improvement Using Simulation and Belief Networks. The reason this problem was used, in the absence of a more complex model, was that its results were readily comparable to that of McCabe (1997) and Carmicheal (1987). The simulation model developed in Chapter 3 and the belief network developed in Chapter 5 were the input files used for the resource allocation improvement.

This chapter is broken down in the following manner: section 6.2 gives an overview to the Visual Basic interface and how one navigates through it; section 6.3 outlines the validation of the model; and section 6.4 concludes the chapter.

6.2 Overview of the Visual Basic Interface

The Visual Basic interface can be considered the control center for the resource allocation improvement process. The interface is where the user specifies which simulation model and belief network is to be used for the resource allocation improvement process. The following sub sections outline the steps involved in operating the interface and take the reader through the validation example.

6.2.1 Linking a Belief Network to a Simulation Model

The linking of the belief network to the simulation model is done via the Visual Basic Interface. When running the resource allocation improvement process from the interface, the first form the user is given is the Resource Allocation Improvement Routine form (Figure 6.1) where the user selects the simulation and the belief network file. Both files must be created in their respective programs before running the interface. For the validation example, the McCabeTest.sm file, created in Chapter 3, will be the simulation file. The file is selected by navigating through the simulation file directory tree and selecting it from the file list box. When the desired file is clicked, its name appears under the file list box indicating that it is the current file. If the user wishes to change files, they must simply click on another file in the list box. The same process is followed to select the belief network file. For the validation example, the desired belief network file was the McCabe.Dsc file.

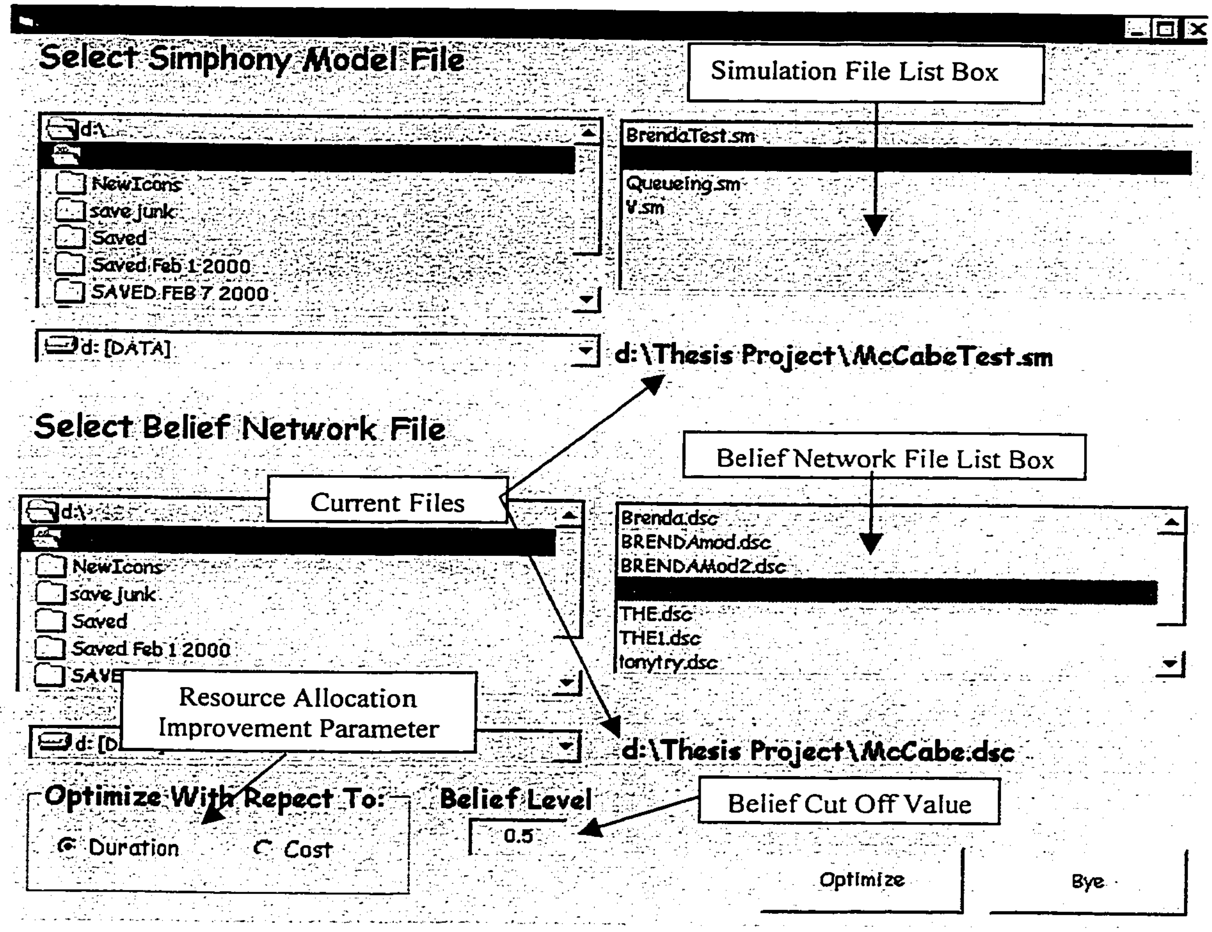


Figure 6.1 Resource Allocation Improvement Routine Form

6.2.2 Initialization Parameters

The parameters that were defined for the resource allocation improvement process were the belief cut off value, the resource allocation improvement parameter, and the linking of the recommendations. The belief cut off value is the value that any recommendation's associated belief must exceed to be considered valid. By default the belief cut off value is set at 0.5. For the validation example, the default value was used.

The resource allocation improvement parameter was defined by clicking the option button for either cost or duration. By default, the selected option is Cost. For the validation example, runs were completed with the resource allocation improvement option set to both cost and duration. Carmicheal (1987) and McCabe (1997) both improved resource allocation with respect to cost, but McCabe also improved resource allocation with respect to project duration. Comparisons were made for both cost and duration.

The linking of the recommendations was done in the Output Recommendations form (Figure 6.2). This form presents the names of each output node and then prompts the user to select a recommendation from a list box. After this is done, the Output Nodes form asks the user to verify the choice of recommendations for the output nodes (Figure 6.3). Upon agreement, the program proceeds with the first iteration of the resource allocation improvement process. If the user disagrees with the selection of recommendations for output nodes, the user can click the Disagree button on the Output Nodes form and the program will return to the Resource Allocation Improvement Routine form. For this example, the "Too Many Servers" node was linked to the "Decrease Server Number" recommendation, the "Too Few Servers" node was linked to the "Increase Server Number" recommendation, the "Too Many Customers" node was linked to the "Decrease Customer Number" recommendation, and the "Too Few Customers" node was linked to the "Increase Customer Number" recommendation.

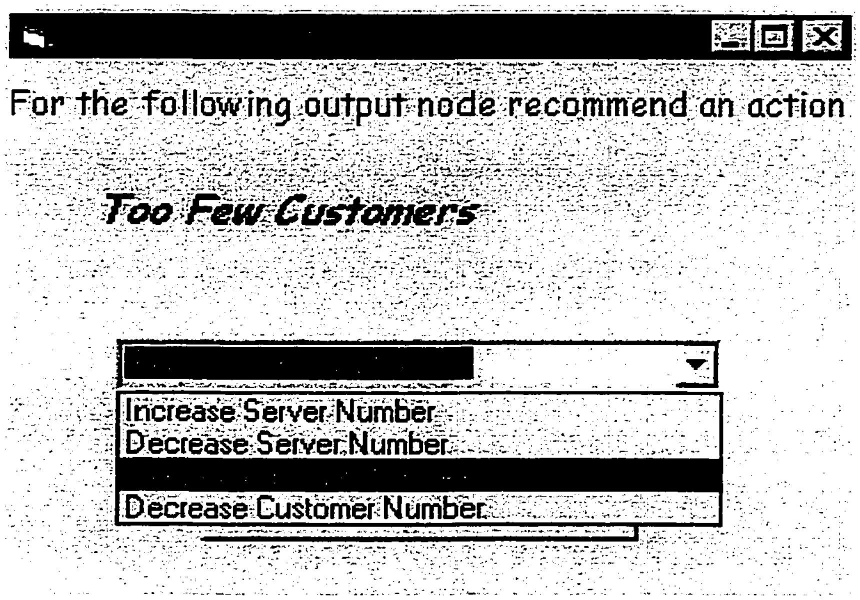


Figure 6.2 Output Recommendations Form

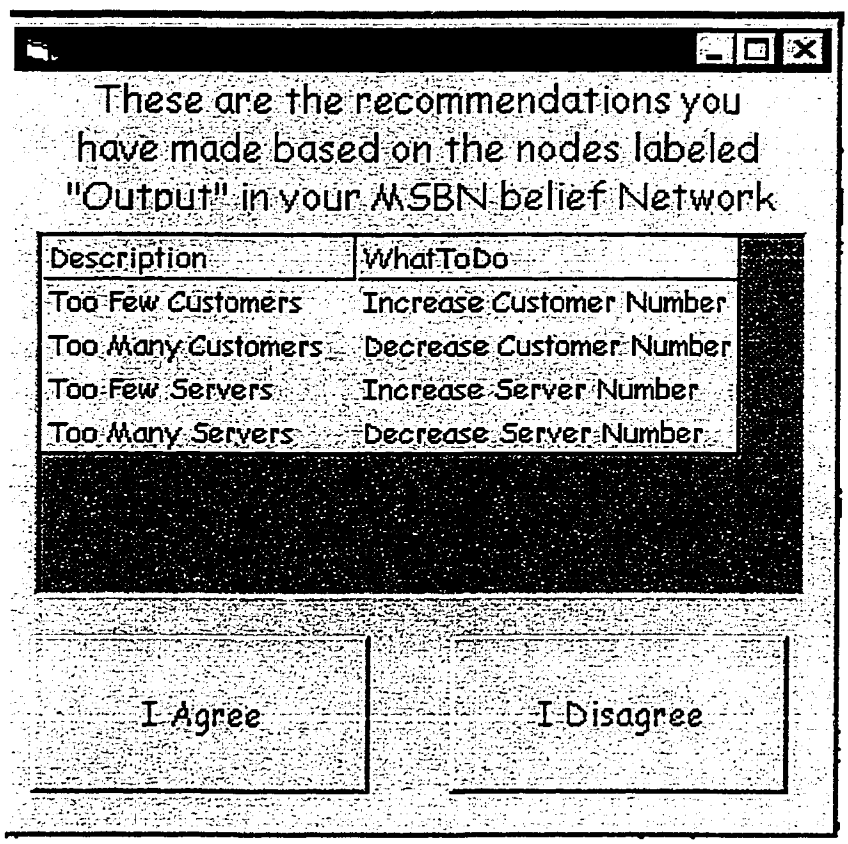


Figure 6.3 Output Nodes Form

6.2.3 Progressing Through the Iterations

Upon completion of the first iteration the user is presented with the Iteration Results form (Figure 6.4), along with the list of viable recommendations. Each recommendation is listed with the Element Identification Number for the element where the problem was experienced, the problem identified, the recommendation for the problem, and the associated belief. By default, the recommendation with the highest belief is listed in the Action to Be Taken grid. The user has the ability to accept the action, or, by clicking one of the other listed recommendations, to replace it with another. If the user was not satisfied with any of the recommendations, they can press the Done button and the resource allocation improvement process will terminate. If the user is satisfied with the selected recommendation they can either click the Iterate button and progress on to the next iteration, or click the "Don't Verify - Have the Computer Automate Process" check box before pressing the Iterate button. If the latter is chosen, the user gives control of the resource allocation improvement process to the computer and does not have to verify the action to be taken for each iteration as the computer progresses through the iterations by taking the recommendation with the highest belief for each iteration. When automating the process, the user must specify the maximum number of iterations. For the validation example thirty iterations were specified as the maximum.

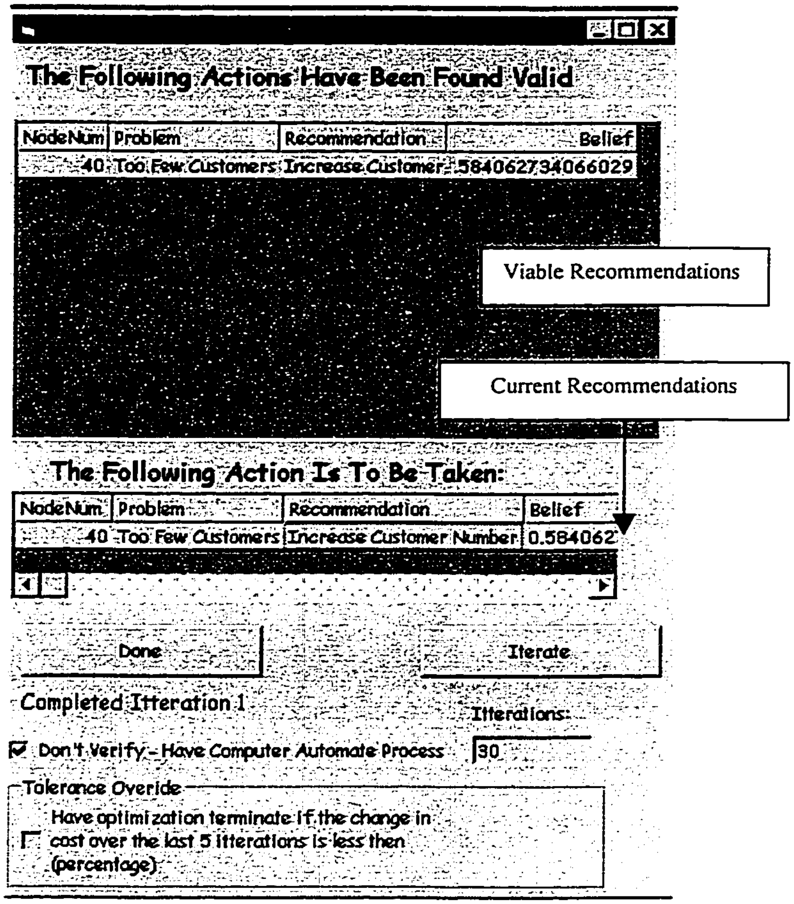


Figure 6.4 Iteration Results Form

When the user clicks the automation check box, the user is presented with another check box that deals with a tolerance override for the resource allocation improvement process. What this does is allow the user to terminate the process if the change in the cost or duration, whichever is specified for the resource allocation improvement parameter, is below a user-specified value over the last five iterations. This feature was not used for this example.

6.2.4 Resource Allocation Improvement Termination

All actions taken in the resource allocation improvement process are recorded in an Access database file. This allows the user to view what occurred during the resource allocation improvement process. The RunResults table allows the user to see, for each iteration, at which element the problem was being experienced, the problem identified, the recommendation for the problem, the belief value of the problem, the project cost, the project duration, and the elements that were changed. The last record in the table is empty except for an entry in the Recommendations field, which explains why the resource allocation improvement process was terminated. Terminations can occur due to a lack of further recommendations, the maximum iterations being reached, or due to the breaching the tolerance value. Both resource allocation improvement runs of the validating example were terminated due to no further recommendations being listed.

6.3 Model Validation

The simulation model was run each time for ten iterations before outputting the performance indices to the Visual Basic interface. The following data was taken from one of the iterations using four trucks. The server utilization was found to be 0.48, while Carmicheal had determined a utilization of 0.49, resulting in an error of 2.04%. The backcycle time is determined by subtracting the service time from the cycle time. The cycle time was found to be 0.26 and the service time was found to be 0.03, therefore the simulated backcycle time was calculated at 0.23. Carmicheal specified a backcycle time of 0.2222 hours (4.5 trucks per hour) and a service time of 0.032 hours (31.1 trucks per hour). The degree of error for the backcycle and service times were -3.51% and 6.25 % respectively. Thus, with only these small errors, the model appears to be simulating adequately.

Upon inspection of the database file for the cost resource allocation improvement, it was found that the preferred number of trucks for this operation was 6. This is in agreement with the results obtained from both Carmicheal and McCabe. This preferred allocation did not meet all the constraints supplied by the user, as the termination of the simulation was due to no further recommendations. Upon inspecting FullEvalTable in the database file, it was found that the last belief network evaluation had a recommendation from Element # 40 to increase the customer number (Table 6.1). This indicates that one or several of the performance indices were outside of their permissible range. The recommendation was rejected as it was not possible to increase the number of customers past its maximum allowable limit.

Element	Problem	Recommendation	Belief
50	Too Few Customers	Increase Customer Number	0.1215
50	Too Many Customers	Decrease Customer Number	0.1739
50	Too Few Servers	Increase Server Number	0.2291
50	Too Many Servers	Decrease Server Number	0.3605
40	Too Few Customers	Increase Customer Number	0.5350
40	Too Many Customers	Decrease Customer Number	0.1272
40	Too Few Servers	Increase Server Number	0.3693
40	Too Many Servers	Decrease Server Number	0.6188

Table 6.1 Belief Network Evaluation for Last Iteration of Cost Improvement

From inspection of the RunResults and RunResultsSettings table in the database, it was found that the preferred cost of $2,467.03 for the project occurred in iteration 3 with 6 trucks. It should be noted that the iterations following iteration 3 did make changes to the model to try improve the queuing characteristics of the simulation, however, the succeeding changes did not result in a lower cost. Whether improving the model for cost or duration, the changes to the simulation model are based on the user-defined tolerances for the queuing characteristics in the simulation model. When observing the model, the belief network is only concerned with its queuing characteristics and is not concerned about the total project cost or duration until the completion of the resource allocation improvement process when the system searches the database files to find the iteration with the lowest cost. Once this iteration is found, the computer sets the simulation model configuration to the resource allocation corresponding to that iteration. Thus, even though the lowest project cost was found in iteration 3, the computer carried on with succeeding iterations to see if the recommendations from the belief network would result in a better project cost. This was not the case for this model, and iteration 3 was found to be the preferred resource allocation scenario. This was confirmed when checking the resource allocations in the simulation model at the completion of the resource allocation improvement. The settings in the simulation model were those of iteration 3 with 6 trucks. The costs, durations, and resource allocations for each iteration can be found in Table 6.2.

Iteration	Element	Problem	Recommendation	Belief	Trucks	Cost	Duration
1	40	Too Few Customers	Increase Customer Number	0.5841	4	$2,587.83	7.201
2	40	Too Few Customers	Increase Customer Number	0.5841	5	$2,498.24	5.958
3	40	Too Few Customers	Increase Customer Number	0.5350	6	$2,467.03	5.160
4	40	Too Few Customers	Increase Customer Number	0.5350	7	$2,477.97	4.609
5	40	Too Few Customers	Increase Customer Number	0.5350	8	$2,577.97	4.317
6	40	Too Few Customers	Increase Customer Number	0.5350	9	$2,663.21	4.056
7	40	Too Few Customers	Increase Customer Number	0.5350	10	$2,747.42	3.837

Table 6.2 Summary of Results for Cost Resource Allocation Improvement

The termination of the duration resource allocation improvement occurred in a manner similar to that of the cost resource allocation improvement. Inspection of the database file shows that there was a final recommendation from Element # 40 to increase the customer number (Table 6.3). Again, this recommendation was not considered valid because the customer number had reached its maximum allocation limit. This again indicates that one or several of the performance indices were not satisfied at the completion of the resource allocation improvement.

Element	Problem	Recommendation	Belief
50	Too Few Customers	Increase Customer Number	0.1737
50	Too Many Customers	Decrease Customer Number	0.2097
50	Too Few Servers	Increase Server Number	0.2223
50	Too Many Servers	Decrease Server Number	0.2413
40	Too Few Customers	Increase Customer Number	0.6920
40	Too Many Customers	Decrease Customer Number	0.0904
40	Too Few Servers	Increase Server Number	0.3374
40	Too Many Servers	Decrease Server Number	0.4620

Table 6.3 Belief Network Evaluation for Last Duration Improvement Iteration

Upon checking the resource allocation in the simulation model, it was found that iteration 7 with 10 trucks provided the preferred project duration of 3.837 hours. This is in agreement with McCabe's findings. A summary of the data in the database tables for the duration resource allocation improvement can be found in Table 6.4.

Iteration	Element	Problem	Recommendation	Belief	Trucks	Cost	Duration
1	40	Too Few Customers	Increase Customer Number	0.6336017	4	$ 2,587.83	7.201
2	40	Too Few Customers	Increase Customer Number	0.6336017	5	$ 2,498.24	5.958
3	40	Too Few Customers	Increase Customer Number	0.69197073	6	$ 2,467.03	5.160
4	40	Too Few Customers	Increase Customer Number	0.69197073	7	$ 2,477.97	4.609
5	40	Too Few Customers	Increase Customer Number	0.69197073	8	$ 2,577.97	4.317
6	40	Too Few Customers	Increase Customer Number	0.69197073	9	$ 2,663.21	4.056
7	40	Too Few Customers	Increase Customer Number	0.69197073	10	$ 2,747.42	3.837

Table 6.4 Summary of Results from Duration Resource Allocation Improvement

6.4 Conclusion

This chapter introduced the reader to the Visual Basic interface and a simple queuing system was used to demonstrate and validate the model. The inputs for the system were the simulation file and belief network created in chapters 3 and 5 respectively. Output from the system was an improved simulation model with respect to its resource allocation.

The interface has been designed with simplicity in mind. There are four forms the user is presented with. Inputs from these forms are kept to a minimum to avoid user error and confusion. The first form the user is presented with was the Resource Allocation Improvement Routine form, where the user selects the simulation and belief network files, defines the belief cut off value, and indicates whether resource allocation improvement should take place with respect to cost or duration. The next form is the Output Recommendations form, where the user selects recommendations from a list box in order to link them to output nodes in the belief network. The next form the user is presented with is the Output Nodes form, which lets the user verify their selection of recommendations for output nodes. The last form is the Iteration Results form, where the user views the recommendations for each iteration and oversees the resource allocation improvement process. Here the user has the ability to manually select each recommendation for each iteration or let the computer automate the process.

Lastly, the results from the validation example proved the system to be valid. The simulation template accurately simulated the system with errors of 2.04%, -3.51%, and 6.25 % for the server utilization, backcycle time, and service time. The resource allocation improvement routine successfully found that 6 trucks result in the preferred cost while 10 trucks results in the preferred duration.

Recommendations to improve this interface would deal with developing a better method of presenting the user information upon termination of the resource allocation improvement. A possible fifth form could be developed in Visual Basic that presented the information in the database to the user, highlighting the preferred resource allocation for their simulation model. This would prevent the user from having to search the database file when inquiring about events occurring during the resource allocation improvement process. Secondly, information could be provided as to which user defined constraints were not met for each iteration. This would allow the user to understand what is occurring in the resource allocation for each iteration.

Chapter 7

Conclusion and Recommendations

7.1 Conclusions

The scope of this research was two-fold. The first objective was to develop a descriptive modeling template that was able to capture and describe the construction process being modeled. The second objective was to develop a method for improving the resource allocation in the simulation model.

The descriptive simulation template developed in this research was based on the CYCLONE simulation modeling system developed by Halpin and Riggs (1992). The descriptive modeling process was solved by identifying the activity modeling elements used in the model and creating a modeling structure for the user to follow when modeling, developing a structured order for the elements in the model. This structure was provided via a loop and an element identification system. The loops identified the individual cyclic activities within the model, and the Element Identification numbers were used to assign the position of the elements within these loops.

Resources for the activities were organized through the relationships between the Combi and Queue elements. A fundamental rule of CYCLONE is that a Queue Element must precede a Combi Element. Thus for each Combi node the required resources for that activity were found in the preceding Queue Element(s).

All information from the descriptive modeling process is presented in two tables, the Activity Table and the Resource List Table. The Activity Table describes the construction processes occurring in the model and describes to the user the activities occurring in the model, the order the activities occurred in, and the resources required for the activities. The Resource List Table identifies the resources required for the model, the activities they were required for, and what additional resources were required to carry out the activity.

The second objective of this research, the resource allocation improvement process, was solved using performance indices and a belief network. The performance indices communicated the queuing states, server utilizations, and customer delays occurring in the simulation model to the belief network. The belief network acted as an expert system by taking the performance indices and identifying possible problems in the resource allocations for the current simulation run, which were then linked to recommendations for improving to the simulation model. The system is set up such that it is generic in nature and views the resources in the model as either servers or customers, as in queuing theory. Therefore the recommendations for model improvement are limited to increasing or decreasing the number of servers or customers.

The performance indices used to describe the simulation model to the belief network are the server utilization, customer delay, average queue length, and average queue wait indices, which are calculated at every queuing situation in the model. The simulation modeling queue elements intrinsically calculate these indices, the Customer Queue Elements calculate the customer delay, average queue length, and average queue wait, and the Server Queue Elements calculate the server utilization, average queue length, and average queue wait.

The user develops the belief network for use with this system thus able to create any type of belief network they desire and define how it interprets the indices and influences the simulation model. The user must develop the network according to the manner described by the author, which means that the input and output nodes must be labeled as requested, otherwise the system will be unable to identify them.

The system uses using an iterative process of simulating the model, sending the indices to the belief network, having the belief network identify problems in the model, and implementing changes to the model. Upon completion of this process, the database file, where all the actions from the system are recorded, is called upon. At the completion of the iterative process, a search of the database occurs to find the iteration with the lowest cost or duration, depending on what constraint is specified for the resource allocation improvement. Once the preferred iteration is found, the model is set to the resource allocations for that iteration.

The mechanism that links everything together is a Visual Basic interface. The user creates the simulation model and belief network outside of this interface. In the interface the user selects the desired simulation model and belief network file, then defines some properties for the resource allocation improvement process to follow. The rest is left to the computer. Throughout the development of this prototype model, simplicity and error handlers have been stressed to minimize the chance of user error and confusion.

7.2 Contributions

The first contribution of this research is the development of a descriptive modeling system. The problem identified in the current manner of project management was a gap in communication as the project was passed on from designer to estimator to field construction personnel. Generally speaking, the construction industry is described as a "Trade" or "Craft" culture where the construction methods in a project are determined by field personnel. This usually occurs in an unstructured manner with little in the way of a scientific process. There was therefore a breakdown identified in the planning process that led to duplication of effort, a modification of the original construction plan, or even omissions in some of the details of the construction plan. The descriptive modeling process provides a means of communicating the construction plan to all parties involved. Traditionally the only pieces of information that are passed on between parties are the construction drawings and plans. This tool acts as the equivalent of the drawings and specifications while also facilitating the communication of the construction method. It therefore provides a means of letting the designer, estimator, and field construction personnel know how the project is to be undertaken. The benefits from this medium are a reduction in errors occurring during construction, less work having to be redone, increased construction productivity, reduced project costs, and improved communication leading to less frustration and greater satisfaction for all parties involved.

The second contribution of this research is a method of improving resource allocation for a project. Construction projects are considered a unique, one time only undertaking. This allows a constructor only one attempt to successfully carry out the project. The use of simulation allows one to conceptually carry out a project many times over, before actual construction, with minimal cost until a favored plan of attack is found. The resource allocation improvement process described here works cooperatively with simulation to define this favored plan of attack. The use of a belief network aids in recommending improvements to the simulation model to define a preferred allocation of resources for the project. An expert defines the belief network, which allows the user the ability to control how the belief network will make recommendations and influence the resource allocations for the project. This resource allocation improvement process gives the user a means of directing the changes to the simulation model, direction in improving the resource allocations for the model, and a means of defining the favored plan of attack for the project.

The end result of this research is a scientific method that effectively defines and communicates the most efficient method of constructing a project. The descriptive modeling process defines what is to be done in the project while the resource allocation improvement process defines what is required for it to be done. By using this system from the conceptual stage of a project on through to the project completion, assurance can be made as to the identification and communication of the construction method between all parties involved, resulting in a common focus on one construction plan. There would be no duplication of effort in redefining the construction method as the project is passed from the designer to the estimator to the field construction personnel. Benefits from this system would lead to less deviation from the original construction plan, reduced chance of errors or omissions during construction, and minimize project costs.

7.3 Recommendations

The only recommendation the author can think of for the descriptive portion of this research is with regard to the aesthetics of the outputs. Though the current method can clearly describe to the user the construction process and the resources required, some users may want the outputs customized to their liking or find some information unnecessary. Other users may find all the information vital, but may wish it to be presented in a different manner. For the moment, the tables only present static information about the simulation model and how the model is drawn. Further work can be done to include the dynamic information about the model obtained from the simulation itself. Further improvement on the aesthetics of the outputs could include the loss of the table format and their replacement with a text dialogue describing the simulation model. Further work could also be done to allow additional input information to be incorporated into the system thereby more fully capturing the specifications and scope of the project being modeled.

A recommendation to the simulation template worthy of mention is the further development of the elements to allow for infinite connection points for each element. Due to the complexity of this problem and the time required for its solution, the prototype model was developed with each modeling element being able to have a maximum of only four connection points.

For the resource allocation improvement process, the author can find five major recommendations. The first of these involves further work on the template by creating two additional modeling elements used to calculate the performance indices of non-cyclic activities. One of these elements would create the entity and the other would read the entity's information as it leaves the simulation model. This would extend the functionality of this system to problems of a non-cyclic nature, such as a gravel crushing operation.

The second recommendation deals with the Queue node modeling elements. Currently, there are two Queue Elements that represent customers and servers. There is a problem with this set up, as some entities may take on duel roles as both customers and servers. As explained earlier, a loader can be considered a sever for incoming trucks and a customer for an equipment maintenance crew. A possible solution to this problem is to create a generic queue element that represents both customers and servers and have the Combi element perform the evaluation of the queuing characteristics of the simulation. This would involve redefining both the Customer Delay index and the Server Utilization index into one single index.

The third recommendation is that further work be done to create a system that would allow the user to define the performance indices and input nodes for the belief network. Rather than simply having the four performance indices used in this research to describe the simulation model to the belief network, it would prove beneficial if the user could define their own performance indices for their model. This would give the user the ability to create a more domain specific system for their needs while still maintaining a system generic in nature. The user could then create specialized performance indices for describing a dewatering simulation model, a gravel crushing simulation model, or a tunnel boring simulation model. This would also require further development to allow the user to define the input nodes for the belief network. This would allow the user to have input nodes that could describe the characteristics of the user-defined performance indices. These nodes could be used to identify if a shovel bucket size is too large or too small, or if a screen for a gravel crusher should be increased or decreased.

The fourth recommendation found by the author addresses the selection of the recommendations for the model improvement. The author finds two main areas of improvement for the selection process. Firstly, further work could be done to develop a method in which the recommendation with the highest belief is chosen regardless of any conflicts this recommendation may have with other recommendations. Secondly, further work could be done to develop a method in which combinations of recommendations could be evaluated and implemented for a given iteration.

The author's last recommendation for the resource allocation improvement portion of this system is the further development of the presentation of information at the end of the resource allocation improvement process. Currently, the resource allocation improvement process terminates, changes the simulation model to the preferred resource allocation, and records all its actions in the database. The development of another form for the interface could prevent the user from having to search the database to understand what occurred in the resource allocation improvement. This form could provide the user with a summary of the results for each iteration and give details as to which constraints were not met for each iteration.

REFERENCES

References

AbouRizk, S. M. and Shi, J. "Automated Construction-simulation Optimization" Journal of Construction Engineering and Management 120.2 (1994): 374-385

Ahuja, Hira N., Dozzi, S. P., and AbouRizk, S. M. (1994) Project Management Techniques in Planning and Controlling Construction Projects, John Wiley & Sons Inc., New York, New York

Al-Tabtabai, H., Diekmann, J. E. "Knowledge-Based approach to construction project control" International Journal Of Project Management 10.1 (1992): 23-30

Alkass, Sabah and Harris, Frank "Expert System For Earthmoving Equipment Selection In Road Construction" Journal Of Construction Engineering and Management 114.3 (1988): 426-440

Amirkhanian, Serji N. and Baker, Nancy J. "Expert System For Equipment Selection For Earthmoving Operations" Journal Of Construction Engineering and Management 118.2 (1992): 318-331

Carmicheal, D. G. Engineering Queues in Construction and Mining. Chichester: Ellis Horwood Limited, 1987.

Charniak, Eugene "Bayesian Networks without Tears." AI Magazine 12.4 (1991): 50-63

Chong, H.G. and Walley, W. J. "Rule Based versus probabilistic approaches to the diagnosis of faults in wastewater treatment processes", Artificial Intelligence in Engineering, Vol. 10.3 (1996): 265-273

Davidson J. M. "An Application of Expert Systems Technology in Process Engineering" Instrumentation in the Chemical and Petroleum Industries Volume 20. Instrument Society of America, 1988 p.85 - 91

Der Kiureghian, Armen "Bayesian Analysis of Model Uncertainty in Structural Reliability Methods" Construction Congress V: managing engineered construction in expanding global markets. Reston, VA: American Society of Civil Engineers, 1997. p. 211 - 221

Doukidis, Georgios I. And Paul, Ray J. "Research into Expert Systems to Aid Simulation Model Formulation" Journal of The Operational Research Society. Great Britain: Operational Research Society, 1985. p. 319 - 325

El-Choum, Mohamed K., and Rumala, Frank "A Conceptual Model For Controlling Underground Construction Projects" Lecture Notes in Engineering: Reliability and Optimization of Structural Systems '90. Berkeley, California: International Federation for Information Processing Work Group, 1990. p. 338 - 347

Halpin, Daniel W. and Riggs, Leland S. (1992) Planning and Analysis of Construction Operations, John Wiley & Sons, Inc., New York

Helweg Otto, J. and El-Khashab Ahmed "Proposal for Combining Expert Systems and Canal Design" National Water Conference. New York, New York: American Society of Civil Engineers, 1989. p. 159 - 163

Hendrickson, Chris T., Martinelli, David, and Rehak, Daniel "Heirarchial Rule-Based Activity Duration Estimation" Journal Of Construction Engineering and Management 113.2 (1987): 288-301

Hendrickson, Chris T., et al. "Expert System For Construction Planning" Journal Of Construction Engineering and Management 1.4 (1987): 253-269

Kartam, Nabil A. "ISICAD: Interactive System for Integrating CAD and Computer-Based Construction Systems" MicroComputers in Civil Engineering 9.1 (1994): p.41 - 51

Kriviak, Gary J. and Scanlon, Andrew "Bayesian Analysis of In-situ Test Data for Estimating the Compressive Strength of Concrete in Existing Structures" University of Alberta Department of Civil Engineering Structural Report n 140 (July 1986): 161 pages

McCabe, Brenda Yvette "An automated Modeling Approach for Construction Performance Improvement Using Simulation and Belief Networks." Doctor of Philosophy Thesis, University of Alberta, 1997

McGartland, Martin R., and Hendrickson, Chris T. "Expert System For Construction Project Monitoring" Journal Of Construction Engineering and Management 111.3 (1986): 293-307

Minkarah, Issam, and Ahmad, Irtishad "Expert Systems As Construction Management Tools" Journal Of Management Engineering 5.2 (1989): 155-163

Navon, Ronie "CAM/CAD Based Automatic Production and Construction-Data Extraction" Proceedings of the 1995 Construction Congress. New York, New York: American Society of Civil Engineers, 1995. p. 347 - 354

Papamicheal K. M. and Selkowitz S. E. "Modeling the Building Design Process and Expertise" ASHRAE Transactions. American Society of Heating, Refrigerating, and Air Conditioning Engineers, INC. 1990 p. 481-489

Paulson, B. C. Jr., Chan W. T., and Koo, C.C. "Construction Operations Simulation by MicroComputer" Journal of Construction Engineering and Management 113.2 (1987): 302-314

Reinschmidt, Kenneth F., Griffs, F. H., and Bronner, Patrick L. "Integration of Engineering, Design, and Construction" Journal of Computing in Civil Engineering 117.4 (1991): p.756 - 772

Russell, Stuart and Novig, Peter (1994) Artificial Intelligence a Modern Approach, Prentice Hall, Inc., Englewood Cliffs, New Jersey 07632

Sathi, Arvind, Morton, Tomas E., and Roth Steven F. "Callisto: An Intelligent Project Management." AI Magazine 7.5 (1986): 34-52

Sorensen J. D. and Engelund S. "Optimal Planning of Maintenance of Concrete Structures" Optimal Performance of Civil Infrastructure Systems. Reston, Virginia: Structural Engineering Institute and American Society of Civil Engineers, 1997. p. 169 - 180

Tommelein, Iris D., Odeh, Abdalla M., and Carr, Robert I. "Knowledge Based Assembly of Simulation Networks Using Construction Designs, Plans, and Methods" Proceedings of the 1994 Winter Simulation Conference 1994 p.1145 - 1152

Tommelein, I. D., et al "SightPlan Experiments: Alternative Strategies for Site Layout Design" Journal of Computing in Civil Engineering 5.1 (1991): p.42 - 63

Trego, Linda. "Computers in Manufacturing" Automotive Engineering May 1997: p. 111-115

Teicholz, Paul and Fischer, Martin "Strategy For Computer Integrated Construction Technology" Journal of Construction Engineering and Management 120.1 (1994): p.117 - 131

Vanegas, Jorge A., Bravo, Edmundo B., and Halpin, Daniel W. "Simulation Technologies for Planning Heavy Construction Processes" Journal of Construction Engineering and Management 119.1 (1993) p. 336 - 354

Varis, Olli. "A belief network approach to optimization and parameter estimation: application to resource and environmental management." Artificial Intelligence 101 (1998): 135-163

Vaugn, Frank. "3D & 4D CAD Modeling on Commercial Design-Build Projects." Computing in Civil Engineering. New York, New York: American Society of Civil Engineers, 1996. p. 390 - 396

APPENDIX A

CONDITIONAL PROBABILITY VALUES

Duration

				Conditional Values	
Parent 1 - Too Many Customers	Parent 2 - Too Few Customers	Parent 3 - Too Many Servers	Parent 4 - Too Few Servers	Good	Optimize
TRUE	N/A	TRUE	N/A	0.55	0.45
TRUE	N/A	FALSE	TRUE	0.55	0.45
TRUE	N/A	FALSE	FALSE	0.40	0.60
FALSE	TRUE	TRUE	N/A	0.45	0.55
FALSE	TRUE	FALSE	TRUE	0.45	0.55
FALSE	TRUE	FALSE	FALSE	0.40	0.60
TRUE	FALSE	TRUE	N/A	0.55	0.45
FALSE	FALSE	FALSE	TRUE	0.40	0.60
FALSE	FALSE	FALSE	FALSE	0.50	0.50

Project Cost

				Conditional Values	
Parent 1 - Too Many Customers	Parent 2 - Too Few Customers	Parent 3 - Too Many Servers	Parent 4 - Too Few Servers	Good	Optimize
TRUE	N/A	TRUE	N/A	0.45	0.55
TRUE	N/A	FALSE	TRUE	0.45	0.55
TRUE	N/A	FALSE	FALSE	0.60	0.40
FALSE	TRUE	TRUE	N/A	0.50	0.50
FALSE	TRUE	FALSE	TRUE	0.55	0.45
FALSE	TRUE	FALSE	FALSE	0.60	0.40
TRUE	FALSE	TRUE	N/A	0.45	0.55
FALSE	FALSE	FALSE	TRUE	0.60	0.40
FALSE	FALSE	FALSE	FALSE	0.50	0.50

Queue Wait Index

				Conditional Values		
Parent 1 - Too Many Customers	Parent 2 - Too Few Customers	Parent 3 - Too Many Servers	Parent 4 - Too Few Servers	QW Low	QW OK	QW High
TRUE	N/A	TRUE	N/A	0.24	0.38	0.38
TRUE	N/A	FALSE	TRUE	0.04	0.30	0.66
TRUE	N/A	FALSE	FALSE	0.10	0.40	0.50
FALSE	TRUE	TRUE	N/A	0.71	0.26	0.03
FALSE	TRUE	FALSE	TRUE	0.33	0.34	0.33
FALSE	TRUE	FALSE	FALSE	0.66	0.30	0.04
FALSE	FALSE	TRUE	N/A	0.20	0.75	0.05
FALSE	FALSE	FALSE	TRUE	0.10	0.50	0.40
FALSE	FALSE	FALSE	FALSE	0.05	0.90	0.05

Queue Length Index

				Conditional Values		
Parent 1 - Too Many Customers	Parent 2 - Too Few Customers	Parent 3 - Too Many Servers	Parent 4 - Too Few Servers	QL Low	QL OK	QL High
TRUE	N/A	TRUE	N/A	0.24	0.39	0.37
TRUE	N/A	FALSE	TRUE	0.04	0.30	0.66
TRUE	N/A	FALSE	FALSE	0.10	0.40	0.50
FALSE	TRUE	TRUE	N/A	0.71	0.26	0.03
FALSE	TRUE	FALSE	TRUE	0.33	0.34	0.33
FALSE	TRUE	FALSE	FALSE	0.66	0.30	0.04
FALSE	FALSE	TRUE	N/A	0.20	0.75	0.05
FALSE	FALSE	FALSE	TRUE	0.10	0.50	0.40
FALSE	FALSE	FALSE	FALSE	0.05	0.90	0.05

Customer Delay Index

Parent 1 - Too Many Customers	Parent 2 - Too Few Customers	Conditional Values	
		CD OK	CD High
TRUE	N/A	0.30	0.70
FALSE	TRUE	0.85	0.15
FALSE	FALSE	0.60	0.40

Server Utilization Index

Parent 1 - Too Many Servers	Parent 2 - Too Few Servers	Conditional Values		
		SU Low	SU OK	SU High
TRUE	N/A	0.50	0.43	0.07
FALSE	TRUE	0.04	0.45	0.51
FALSE	FALSE	0.15	0.70	0.15

Too Many Customers	Prior Probability
TRUE	0.40
FALSE	0.60

Too Few Customers	Prior Probability
TRUE	0.40
FALSE	0.60

Too Many Servers	Prior Probability
TRUE	0.40
FALSE	0.60

Too Few Servers	Prior Probability
TRUE	0.40
FALSE	0.60

APPENDIX B

BELIEF VALUES FOR VALIDATION EXAMPLE

Cost	Duration	SU	CD	QW	QL	Too Few Customers	Too Many Customers	Too Few Servers	Too Many Servers
Consider	Good	Absent	Good	Low	Low	0.89602	0.05531	0.31648	0.61425
Consider	Good	Absent	Good	Low	OK	0.53465	0.12948	0.36933	0.61981
Consider	Good	Absent	Good	Low	High	0.58557	0.43897	0.56549	0.44785
Consider	Good	Absent	Good	OK	Low	0.53500	0.12721	0.36925	0.61882
Consider	Good	Absent	Good	OK	OK	0.12379	0.10134	0.28061	0.45057
Consider	Good	Absent	Good	OK	High	0.34264	0.51258	0.57307	0.31364
Consider	Good	Absent	Good	High	Low	0.58406	0.44353	0.56415	0.45234
Consider	Good	Absent	Good	High	OK	0.34327	0.51788	0.57119	0.32110
Consider	Good	Absent	Good	High	High	0.40905	0.74756	0.69676	0.20097
Consider	Good	Absent	High	Low	Low	0.62287	0.39050	0.35053	0.74966
Consider	Good	Absent	High	Low	OK	0.31909	0.46784	0.37401	0.72857
Consider	Good	Absent	High	Low	High	0.40261	0.85547	0.44160	0.63790
Consider	Good	Absent	High	OK	Low	0.31832	0.46281	0.37377	0.72601
Consider	Good	Absent	High	OK	OK	0.13750	0.29753	0.29724	0.47382
Consider	Good	Absent	High	OK	High	0.34585	0.82285	0.51389	0.37257
Consider	Good	Absent	High	High	Low	0.40257	0.85774	0.44094	0.64360
Consider	Good	Absent	High	High	OK	0.34679	0.82592	0.51191	0.38344
Consider	Good	Absent	High	High	High	0.38584	0.93847	0.63028	0.25573
Consider	Good	Low	Absent	Low	Low	0.78624	0.17500	0.35778	0.87665
Consider	Good	Low	Absent	Low	OK	0.39377	0.29523	0.36939	0.89479
Consider	Good	Low	Absent	Low	High	0.42973	0.80643	0.38844	0.88232
Consider	Good	Low	Absent	OK	Low	0.39373	0.29019	0.36917	0.89404
Consider	Good	Low	Absent	OK	OK	0.12089	0.19769	0.32116	0.76835
Consider	Good	Low	Absent	OK	High	0.33480	0.76896	0.37543	0.76387
Consider	Good	Low	Absent	High	Low	0.42915	0.81021	0.38867	0.88462
Consider	Good	Low	Absent	High	OK	0.33687	0.77628	0.37621	0.77135
Consider	Good	Low	Absent	High	High	0.39513	0.95633	0.40709	0.69447
Consider	Good	OK	Absent	Low	Low	0.84918	0.12275	0.27729	0.54773
Consider	Good	OK	Absent	Low	OK	0.47060	0.25138	0.32699	0.57598
Consider	Good	OK	Absent	Low	High	0.49026	0.66693	0.45599	0.49216
Consider	Good	OK	Absent	OK	Low	0.47092	0.24795	0.32666	0.57403
Consider	Good	OK	Absent	OK	OK	0.12146	0.17387	0.22905	0.36053
Consider	Good	OK	Absent	OK	High	0.34191	0.70045	0.47273	0.29606
Consider	Good	OK	Absent	High	Low	0.48927	0.67059	0.45538	0.49774
Consider	Good	OK	Absent	High	OK	0.34264	0.70420	0.47182	0.30487
Consider	Good	OK	Absent	High	High	0.39423	0.88194	0.58920	0.20779
Consider	Good	High	Absent	Low	Low	0.78624	0.17500	0.35778	0.87665
Consider	Good	High	Absent	Low	OK	0.39377	0.29523	0.36939	0.89479
Consider	Good	High	Absent	Low	High	0.42973	0.80643	0.38844	0.88232
Consider	Good	High	Absent	OK	Low	0.39373	0.29019	0.36917	0.89404
Consider	Good	High	Absent	OK	OK	0.12089	0.19769	0.32116	0.76835
Consider	Good	High	Absent	OK	High	0.33480	0.76896	0.37543	0.76387
Consider	Good	High	Absent	High	Low	0.42915	0.81021	0.38867	0.88462
Consider	Good	High	Absent	High	OK	0.33687	0.77628	0.37621	0.77135
Consider	Good	High	Absent	High	High	0.39513	0.95633	0.40709	0.69447
Good	Consider	Absent	Good	Low	Low	0.94970	0.02661	0.26769	0.50318
Good	Consider	Absent	Good	Low	OK	0.69158	0.09160	0.33745	0.46267
Good	Consider	Absent	Good	Low	High	0.63457	0.32040	0.56038	0.27919
Good	Consider	Absent	Good	OK	Low	0.69197	0.09039	0.33737	0.46195
Good	Consider	Absent	Good	OK	OK	0.18636	0.11006	0.29858	0.32003
Good	Consider	Absent	Good	OK	High	0.33859	0.43100	0.56901	0.16954
Good	Consider	Absent	Good	High	Low	0.63360	0.32322	0.55971	0.28218
Good	Consider	Absent	Good	High	OK	0.33893	0.43420	0.56806	0.17420
Good	Consider	Absent	Good	High	High	0.38208	0.62108	0.61449	0.11414
Good	Consider	Absent	High	Low	Low	0.76008	0.24769	0.29563	0.58132
Good	Consider	Absent	High	Low	OK	0.40353	0.42894	0.33846	0.53669
Good	Consider	Absent	High	Low	High	0.40781	0.78495	0.38068	0.41677
Good	Consider	Absent	High	OK	Low	0.40355	0.42535	0.33807	0.53378
Good	Consider	Absent	High	OK	OK	0.16383	0.32909	0.27627	0.32354
Good	Consider	Absent	High	OK	High	0.33110	0.77149	0.40745	0.21489
Good	Consider	Absent	High	High	Low	0.40773	0.78712	0.38088	0.42266
Good	Consider	Absent	High	High	OK	0.33179	0.77378	0.40737	0.22275
Good	Consider	Absent	High	High	High	0.36868	0.88566	0.46218	0.15688
Good	Consider	Low	Absent	Low	Low	0.89002	0.09161	0.32803	0.79979
Good	Consider	Low	Absent	Low	OK	0.55303	0.22996	0.33845	0.79926
Good	Consider	Low	Absent	Low	High	0.48655	0.72072	0.35244	0.73856
Good	Consider	Low	Absent	OK	Low	0.55378	0.22619	0.33815	0.79828
Good	Consider	Low	Absent	OK	OK	0.16820	0.21053	0.28874	0.65645
Good	Consider	Low	Absent	OK	High	0.35099	0.75961	0.31553	0.55972
Good	Consider	Low	Absent	High	Low	0.48526	0.72489	0.35315	0.74247
Good	Consider	Low	Absent	High	OK	0.35211	0.76512	0.31746	0.56981
Good	Consider	Low	Absent	High	High	0.39160	0.92984	0.31493	0.47451

Cost	Duration	SU	CD	QW	QL	Too Few Customers	Too Many Customers	Too Few Servers	Too Many Servers
Good	Consider	OK	Absent	Low	Low	0.92405	0.06371	0.21865	0.41052
Good	Consider	OK	Absent	Low	OK	0.61902	0.20494	0.27802	0.39333
Good	Consider	OK	Absent	Low	High	0.52589	0.57395	0.40148	0.28475
Good	Consider	OK	Absent	OK	Low	0.61955	0.20303	0.27773	0.39187
Good	Consider	OK	Absent	OK	OK	0.17373	0.20968	0.22227	0.24126
Good	Consider	OK	Absent	OK	High	0.34095	0.66447	0.38974	0.15333
Good	Consider	OK	Absent	High	Low	0.52516	0.57643	0.40147	0.28891
Good	Consider	OK	Absent	High	OK	0.34132	0.66661	0.38980	0.15874
Good	Consider	OK	Absent	High	High	0.37863	0.82138	0.42640	0.11077
Good	Consider	High	Absent	Low	Low	0.91403	0.04982	0.36366	0.27480
Good	Consider	High	Absent	Low	OK	0.59980	0.15090	0.53253	0.20997
Good	Consider	High	Absent	Low	High	0.55592	0.32110	0.74250	0.09992
Good	Consider	High	Absent	OK	Low	0.60006	0.14981	0.53270	0.20896
Good	Consider	High	Absent	OK	OK	0.19634	0.18864	0.49849	0.12372
Good	Consider	High	Absent	OK	High	0.31094	0.45823	0.75201	0.04911
Good	Consider	High	Absent	High	Low	0.55560	0.32249	0.74180	0.10176
Good	Consider	High	Absent	High	OK	0.31112	0.45934	0.75128	0.05107
Good	Consider	High	Absent	High	High	0.35690	0.65074	0.78652	0.03221