University of Alberta

Automated Modeling and Optimization for Construction Simulation

By

Jingsheng Shi ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy.**

Department of Civil Engineering

Edmonton, Alberta

Fall 1995

ISBN 0-612-06287-2

University of Alberta

Library Release Form

Name of Author: Jingsheng Shi

Title of Thesis: Automated Modeling and Optimization for Construction Simulation

Degree: Doctor of Philosophy

Year this Degree Granted: 1995

Jingsheng Shi
#845, 11121, 82 Avenue
Edmonton, Alberta
T6G 0T4

DATE: Aug. 10, 95

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 **AUTOMATED MODELING AND OPTIMIZATION FOR CONSTRUCTION SIMULATION** submitted by **JINGSHENG SHI** in partial fulfillment for the degree of **DOCTOR OF PHILOSOPHY**.

Dr. S.M AbouRizk

Prof. S.P. Dozzi

Prof. W.A. Weir

Dr. W.J. Sproule

Dr. J.D. Whittaker

Dr. Y. Beliveau

DATE: June 26, 1995

To My Father, My Mother,

My Wife Junli, and My Daughter Sandy

- For Their Love, Encouragement, and Support

ABSTRACT

This thesis presents the automated process for simulating construction projects in order to simplify simulation and to make it a simple tool for construction practitioners. To substantiate an automated simulation system, this thesis has specifically studied the automated modeling and optimization techniques. The Resource-Based Modeling (RBM) was developed as an automated modeling tool mainly for resource-intensive construction projects. A prototype environment with friendly user interface was implemented for modeling earthmoving operations by using the RBM methodology. It enables the modeler to build an accurate simulation model for an earthmoving project using resources as the basic building blocks. The modeling process is automated with the user specifying resources and site conditions for a given project. The user does not have to be proficient in simulation as required by the current simulation process. Little time is required to construct a simulation model for a project by using this RBM environment. The heuristic simulation optimization method presented can automatically locate an acceptable resource allocation by optimizing the user specified objectives based on resource utilization. This method is generic and can be implemented to any simulation package to automate the simulation optimization process. As an extension to this method, a hybrid method has been studied by combining computer simulation with analytical techniques. This approach can use the advantages and avoid the disadvantages of both simulation and mathematical modeling techniques. The automated modeling and optimization techniques developed in the thesis have solved two key issues toward a fully automated simulation system.

ACKNOWLEDGEMENTS

It has been an excellent experience to have worked under the direction of Professor Simaan M. AbouRizk. I would like to thank him for providing me with good research conditions, challenging guidance, and comprehensive training. I am also very grateful to the other faculty members in the Construction Engineering and Management group at the University of Alberta -- Professor Peter Dozzi and Professor William Weir for their miscellaneous support. I would also like to thank the other members served in my committee -- Professor Ming Zuo and Professor William J. Sproule for their technical support and patience.

I would like to acknowledge North American Construction Group for supporting and facilitating this research. My special thanks goes to Mr. Gordon Parchewsky, Mr. William Koehn, and Mr. Rodney Wales for their support.

I owe special thanks to my wife Junli, and my daughter Sandy for their understanding and sacrifice for me. Without their continuous love and support, I would not be able to complete my Ph.D. work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

THESIS SCOPE AND OBJECTIVES

1.1 INTRODUCTION

Construction operations are characterized by the random nature involved in construction processes and by the dynamic interactions between resources and processes (Paulson et. al. 1987, Shi and AbouRizk 1994). Many traditional analytic methods (e.g. CPM) fail to address these key issues and have shown various limitations in planning and scheduling construction projects. Computer simulation provides advantages in modeling uncertainties and dynamics, and has been proven to be an effective tool for planning and scheduling construction projects (Halpin 1977, Vanegas el. al. 1993).

Computer simulation is defined as the process of designing a mathematical-logical model of a real world system and experimenting with the model on a computer (Pritsker 1986). From the user's prospective, three major phases can be identified in using simulation to resolve a real world problem as shown in Figure 1-1: modeling, experimentation and optimization.

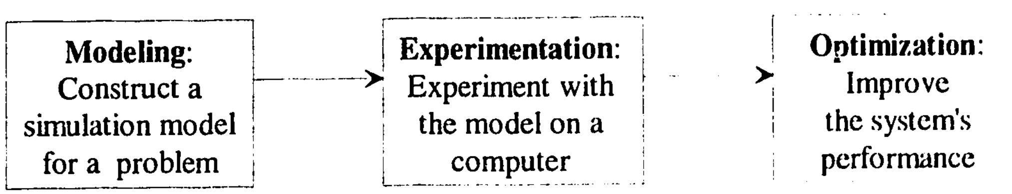


Figure 1-1 Three phases of computer simulation

A model is the description of a real world system. Modeling is a process to describe a stated problem in terms acceptable to a computing system (Pritsker 1986). In this stage, the simulationist is required to understand the real world problem and to represent it in a certain syntax. Experimentation is to execute the simulation model so that the system's operations can be duplicated on a computer and the system's dynamic operating behaviors can be observed. After each experimentation, the user has to analyze the simulation outputs, to change input configuration, and to repeat experimentation until enough simulation scenarios have been conducted and an acceptable solution can be selected through comparison. The simulation optimization, in the context of this thesis, is defined as searching a feasible resource allocation to minimize or maximize the objective of a project.

Currently, the three phases are separately conducted in the process of simulating a real world system. Among the three phases, modeling is the most difficult and time-consuming process because it requires high level of knowledge in both simulation and the real world problem. With the development in simulation studies, simulation experimentation is becoming easier and faster. Many commercial simulation packages can model and simulate very complicated system situations. The manually manipulated optimization phase is another time-consuming process. For a system with 100 possible scenarios, the user has to repeat the experimentation and simulation output analysis process 100 times before the optimum solution can be located.

In the past two decades, construction simulation has been used successfully in academic research with limited successful applications in the industry. One of the major obstacles to achieve wide applications is the complexities involved in constructing a model and the resultant time requirement—the technique is not yet cost effective (Shi and

AbouRizk 1994). A simulation workshop of National Science Foundation concluded that seven of the eleven future research issues must deal with simplifying simulation and making it an operational tool for the construction site (Ibbs 1987, McCahill and Bernold 1993).

From the perspective of a construction project manager, an ideal simulation system should be fully automated. With the user specifying project and resource information, a simulation model should be automatically constructed. A simulation language is then called to experiment with the model. After experimentation, the system should be able to analyze the simulation results, and to search the optimal solution for a given project. This automated simulation process is illustrated in Figure 1-2.

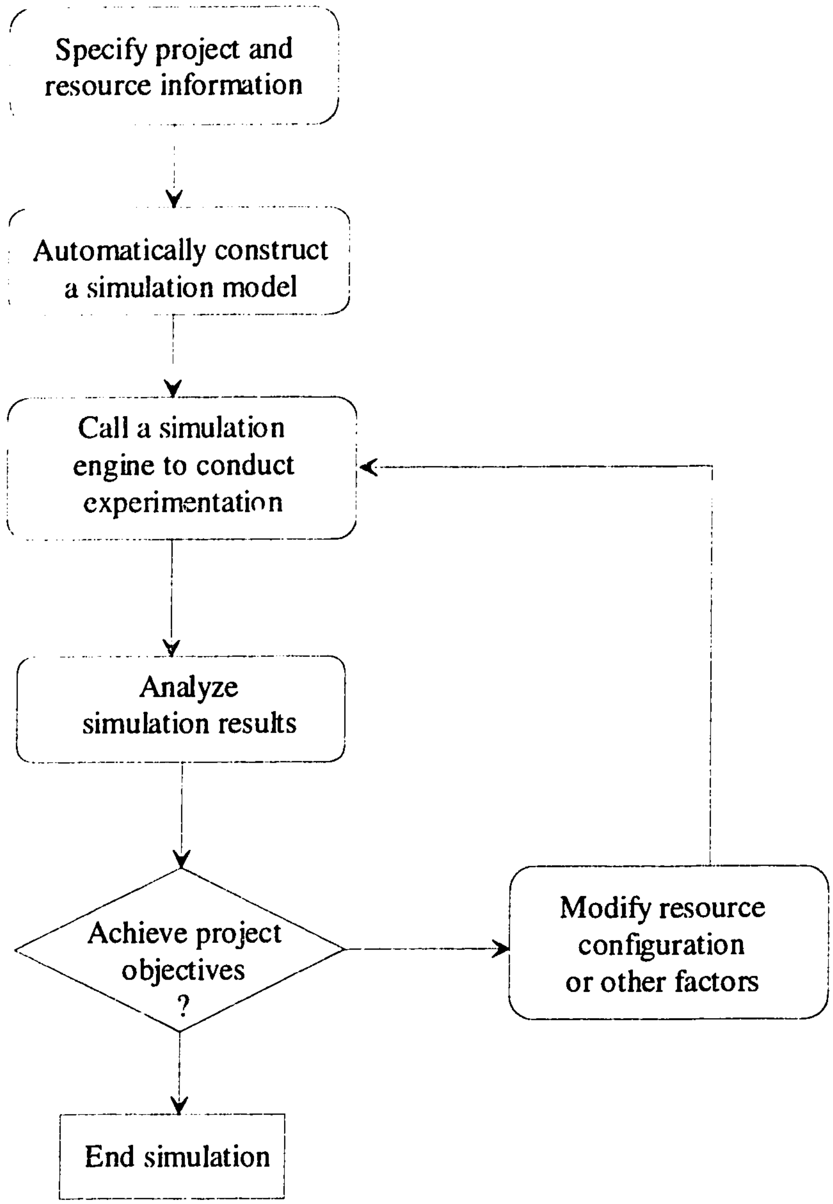


Figure 1-2 An ideal fully automated simulation system

1.2 RESEARCH SCOPE

The major research efforts in both general simulation and construction simulation have been focusing on the development of general purpose simulation languages as detailed in Chapter 2. Many commercial simulation packages are well developed to enable the user to model and simulate complicated real world systems. However, automated modeling, optimization, and automation of the simulation process have not received enough attention because of the difficulties involved. Paul (1992) noted that "it is impossible to produce an all purpose simulation modelling system that can handle any problem that one might wish to model. The analyst is restricted to what a simulation system can handle, or the simulation system must provide programming code that can be modified to do the task that has been set."

In the construction business, most contractors are specialized in conducting some specific types of projects. For example, an earthmoving company mainly does earthmoving-related projects, and a pipeline contractor mainly bids pipeline projects. A framework which can lead to a powerful but simple tool for a specific type of construction projects would be an ideal alternative for a general purpose system in construction simulation. Many advantages can be derived from this approach. First, it is much easier to develop a specific purpose automated simulation system than to develop a general purpose one. The development cost and required time for a specific purpose system would be more affordable. Second, a specific system can be easily integrated with project-type oriented database and knowledge base to further simplify its uses. Furthermore, a specific purpose system will be much simpler for an unsophisticated user to learn how to use it.

Research scope: The research scope of this thesis focuses on the automated modeling and optimization techniques for construction simulation. The modeling methodology can lead to a framework for modeling construction projects. The optimization techniques can assist the user in searching the feasible solution. The two automated phases can be the basis for a fully automated simulation system as illustrated in Figure 1-2.

Most real world construction systems are characterized by dynamic resource interactions, especially in heavy construction. Although each construction system is unique, the operating processes of its component resources are usually somewhat generic. To address the resource-dominating features in construction, the resource-based modeling (RBM) will be studied for automating modeling process. The RBM defines an atomic model library consisting of the operating processes of the resources required for a specific type of construction project. It enables the modeler to build an accurate simulation model for a construction project using resources as the basic building blocks. The modeling process will be automated with the modeler specifying required resources and project information. A prototype RBM environment will be implemented for modeling earthmoving operations as an illustration.

The simulation optimization techniques are to be developed to assist the user in locating optimal resource allocation for a given project. A heuristic method is descried which automates the searching process in locating acceptable resource allocation by optimizing the user selected objective based on resource utilization. As an extension, a hybrid method is presented which combines computer simulation with analytic techniques for optimizing large and complex project systems.

1.3 RESEARCH OBJECTIVE AND ANTICIPATED CONTRIBUTIONS

The objective of this research is to simplify simulation to make it a useful but simple tool to construction practitioners for planning and scheduling construction projects. Using an automated modeling tool, the user does not have to be proficient in simulation as required by current simulation languages. This will enable simulation to be introduced to construction practitioners. An automated simulation system will enable the user to resolve a problem in a cost-effective way by reducing the time required by the simulation process. The anticipated contributions of this research lie in two broad areas: 1) in the academic research of simulation, and 2) in the construction industry;

1. Anticipated academic contributions in simulation research

Computer simulation is still classified as a technique of last resort since it has tended to be expensive and time consuming (Sussman et. al. 1992). The major users of simulation are professional simulationists and researchers at institutions because of the required level of knowledge in simulation. Although education can introduce simulation to more users, an effective alternative would be to simplify simulation and to make it a simple tool that would attract more users to it. An automated simulation system is an ideal approach to achieve that end. The RBM and optimization techniques presented in this thesis will substantiate the automated modeling and optimization processes in simulation. This research will advance simulation studies to another level which addresses the end user (e.g. project managers).

2. Anticipated contributions in the construction industry

Cost-effectiveness and productivity have been the major problems emphasized by the construction industry in the past two decades. Planning and scheduling have been identified as the top potential areas for productivity improvement. Simulation has shown to be a promising tool to resolve some of our planning and scheduling problems. An automated simulation system can be used as easily as traditional analytical techniques but with more powerful functions for modeling uncertainties and dynamic behaviors of construction projects. It allows a project manager to build a simulation model without proficiency in simulation, and enable him/her to directly locate the optimal resource allocation for a construction project. It can overcome some of the major obstacles which are hindering simulation application in construction. Simulation can provide a more accurate and more realistic schedule and plan for a construction project than many analytical techniques do. Cost effectiveness and productivity improvement would be enhanced by the adoption of this research in the high risk and uncertainty construction business.

1.4 THESIS ORGANIZATION

This thesis consists of seven Chapters. Chapter 2 summarizes the state-of-the-art in research related to this thesis. Chapter 3 presents fundamentals for resource-based modeling. An automated modeling system for earthmoving operations is described in Chapter 4. Chapter 5 presents an automated construction simulation optimization technique. As an extension, another simulation optimization method is presented in Chapter 6. Chapter 7 includes conclusions and recommendations for further study.

CHAPTER 2

STATE OF THE ART

2.1 INTRODUCTION

Construction is one of the largest industries in both Canada and the United States. It represents about 10 percent of the gross national product (GNP). According to the Construction Industry Cost Effectiveness Report, productivity fell approximately 20 percent during 1970's in the US (Business Roundtable, 1982). Research has been conducted to identify where the construction productivity can be improved. Nine potential areas for productivity improvement are rated as planning, scheduling, estimating, communication, marketing, procurement, drafting, specifications, and engineering (Choromockos and Mckee 1981). According to a similar survey reported by Arditi (1985), and another one for small to medium size contractors by Koehn and Caplan (1987), planning and scheduling are still the top potential areas for productivity improvement and need immediate research. For the purpose of this thesis, the techniques used for planning and scheduling construction projects are classified into three groups: conventional techniques, simulation-based techniques, and new techniques.

2.2 CONVENTIONAL TECHNIQUES

The conventional techniques are those tools that have been used or are being used in the construction industry. They include bar charts, techniques for planning linear construction, mathematical programming techniques, and network techniques (e.g. CPM).

2.2.1 Bar chart method

For its simplicity, the bar chart method was originally and is still the predominant scheduling method in the construction industry. A bar chart graphically plots activities versus time with the activities being listed vertically. The estimated duration of an activity is plotted as a bar with the start and completion time corresponding to the horizontal time axis. A bar showing actual progress of the activity can be plotted alongside scheduled progress to monitor construction progress. All information including activity description, resource requirements, and other pertinent data can be written on the chart. The main drawback of a bar chart is that it can not detail the logical interrelationships among activities. If an activity is behind schedule, it may not be possible to ascertain the effect of such delay on the completion of the project.

2.2.2 Methods for planning linear construction

Because of the repetitive nature of linear construction projects, some techniques for planning and scheduling purposes have been specially developed, such as 1) Linear Scheduling Method (LSM) (Johnston 1981, Chrzanowski and Johnston 1986), 2) Time Space Scheduling Method (TSSM) (Stradal and Cacha 1982), 3) Velocity Diagram (Roech 1972), 4) Vertical Production Method (VPM) (O'Brien 1975), and 5) Line-Of-Balance (LOB) method.

The first four methods are very similar. They plot lines of constant or varying slopes on two axes to represent activities. One axis is used to represent the time. The other axis is used to record the progress of activities. According to the characteristic of a project, the progress could be distance (for highway and pipeline projects), floor levels (for building construction), or sections (for general). The slope of a line at any position or period of time represents the productivity (or rate of progress) at that location or during that time period. An activity is allowed to have interruptions during the entire length of the project. The diagram is used to plan or record project construction progress on activities which move continuously in sequence along the length of a single project. A typical diagram can be shown in Figure 2-1.

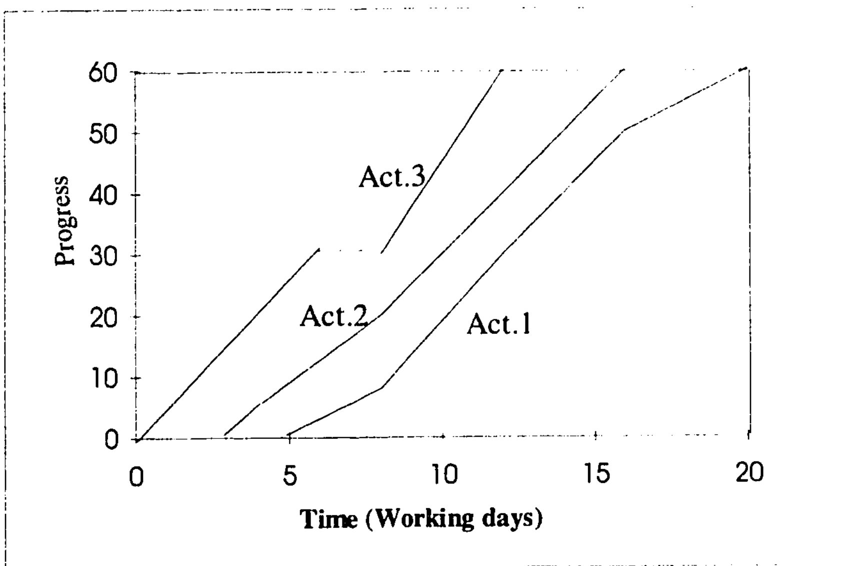


Figure 2-1 A sample LSM diagram

During construction progress of multiple activities, some space buffers including physical distance and time interval between activities may be required. These constraints can be directly represented in the diagram corresponding to axes; i.e. the time constraint (lead or lag) is shown on time axis; and the progress constraints (space buffer) is shown on progress axis. A detailed introduction to LSM and its application in highway construction was presented by Johnston (1981).

The Line-Of-Balance (LOB) technique was developed by the U.S. Navy in the early 1950s. Carr and Meyer (1974) summarized the LOB technique and used it for building construction. Halpin and Woodhead (1976) presented the LOB techniques to high-rise building construction. Arditi and Albulak (1986) adopted LOB to schedule pavement construction. Sarraj (1990) formalized the LOB technique and developed the algorithms without need for any diagram.

In general, the LOB has three diagrams: (1) production diagram; (2) objective diagram; and (3) progress diagram. The production diagram is very similar to a CPM network showing the dependencies among activities and showing the time for one unit completion. Objective diagram is used to show the desired or actual unit completion versus time which is close to a LSM diagram with the difference that LSM diagram shows all activities and LOB objective diagram only shows the unit completion. The progress diagram is plotted unit versus activity similar to bar chart diagram. It shows the progress of each activity in order to achieve the desired objective. The desired and actual progresses are shown in the same chart. The desired LOB line is calculated from production diagram and objective diagram.

2.2.3 Mathematical programming techniques

Mathematical programming techniques, developed with operations research techniques, have been attempted to resolve some planning problems in construction. Mathematical programming techniques require the set up of the objective function(s) and constraint equations by considering resources, site conditions, or other factors. The commonly used objectives in construction are minimizing project duration, minimizing project cost, maximizing resource utilization, or maximizing production rate. Mathematical programming techniques enable the user to optimize a construction plan and schedule. Selinger (1980) uses dynamic programming technique in a bridge project to minimize the total duration by assuming the continuity of resource flowing. Russell and Caselton (1988) extended Selinger's work by using two state variables to allowing interruptions for activities. Moselhi and Ei-Rayes (1993) developed another dynamic model adopting total project cost (including direct construction costs and indirect costs) as the objective. Handa and Barcia (1986) presented a continuous optimal control formulation of a hypothetical highway project. They also used the optimal control theory to solve the same bridge problem from Selinger. Dressler (1974) uses linear programming technique to schedule linear construction by producing a three-dimension time-velocity diagram. Perera (1982) sets a linear programming model to determine the maximum production rate by considering resource sharing in linear construction.

2.2.4 Network techniques

Network techniques were developed in the late 1950's, and were introduced to construction industry in the early 1960's. Network models can overcome the drawbacks of a bar chart by allowing detailing activity interrelationships. Two widely used network models in the construction industry are CPM (Critical Path Method) and PERT (Program Evaluation and Review Technique) for their simplicity comparing with other network models. Many successful researches and applications have been reported (Paulson 1973, 1976, McCough 1982, Jaafari 1984, Russell 1985, Cohenca et. al. 1989, Sidewell and

Cole 1989). CPM is still one of most widely used tools for scheduling, planning and control purposes in the construction industry.

Although network models have many advantages compared with other planning and scheduling methods especially bar chart, they also have some drawbacks. Some researchers have concluded that network techniques are not suitable for linear construction projects. Carr and Meyer (1974) concludes that the line-of -balance method is better than network for repetitive building construction. Cole (1991) concludes that network is best suited for nonrepetitive projects.

One of the main criticisms of traditional network techniques is that they use deterministic approach to estimate activity duration which may lead to optimism. Some researchers use Monte-Carlo based simulation approach to model the uncertainty of activity duration (Carr 1979, Ahuja and Nandakumar 1985, Crandall 1977, and Crandall and Woolery 1982). Monte-Carlo based simulation can overcome the problem in estimating activity duration. However, the dynamic features of construction operations can not be addressed in the traditional networks models.

2.3 SIMULATION TECHNIQUES

Simulation is defined as the mathematical modeling of a system and experimenting with it on a computer (Pritsker 1986). A simulation process involves a Monte-Carlo simulation to model the uncertain or random nature of a system. The dynamic nature of a system is modeled by simulation entities which move through the simulation model during experimentation.

2.3.1 Current State of Knowledge in General Simulation

The early simulation user built up a model for a problem and then implemented a specific computer program to experiment with the model. This process was very expensive and time-consuming. Later on, the general purpose simulation languages were invented such as GPSS and GASP. The user could experiment with different types of simulation models in a general purpose simulation environment. In the early stage of using a general purpose simulation language, the user is required to describe a real world system in an acceptable code format (FORTRAN, C, or others), which then can be experimented. Recently many general purpose simulation languages have been developed (e.g. SLAM II, MODSIM, SES, SIMSCRIPT, SIMAN, GPSS) with powerful and flexible modeling and simulation functions. The graphical modeling functions allow the user to construct a simulation model without accessing lower level programming languages. However, the user still has to be proficient in both simulation theory and the selected simulation language in order to construct a simulation model for a given problem. This modeling process is still very difficult, especially for engineering practitioners.

Simulation researchers have attempted different ways to simplify the modeling process including: 1) Model reusability, 2) computer-aided modeling approach, and 3) hierarchical and modular modeling concepts. Model reusability (Bortscheller and Saulnier 1992) explores the possibility of re-using the models which have been previously created. Computer-aided modeling approach (Balci and Nance 1992, Paul 1992) attempts to construct a simulation model with the user specifying a real world system in a natural language or activity paths. Zeigler (1987) presents the hierarchical and modular modeling concepts: a large complex simulation model is obtained by coupling multiple submodels. If models A and B are submodels with proper modular form, a new model can be created by specifying how the input and output ports of A and B are to be connected to each other and to external ports. Luna (1991, and 1992) uses the concepts of hierarchy and modularity to implement a modeling and simulation environment in an object-oriented language (Smalltalk). The entire simulation model is obtained by combining proper submodels and components. The interface of objects is by message passing from source (the sender) to the receiver. However, Paul (1992) noticed that "it is impossible to produce an all purpose simulation modelling system that can handle any problem that one might wish to model. The analyst is restricted to what a simulation system can handle, or the simulation system must provide programming code that can be modified to do the task that has been set."

2.3.2 Current State of Knowledge in Construction Simulation

Halpin (1973) developed the CYCLONE modeling methodology. It has been the basis for a number of construction simulation systems including INSIGHT (Paulson 1978), RESQUE (Chang 1987), UM-CYCLONE (Ioannou 1989), COOPS (Liu and Ioannou 1992), DISCO (Huang et. al. 1994), CIPROS (Tommelein and Odeh 1994), STROBOSCOPE (Martinez and Ioannou 1994), and HSM (Sawhney and AbouRizk 1995).

In industrial engineering, many simulation systems have been introduced, such as SLAM II, MODSIM, SES, SIMSCRIPT, SIMAN, GPSS, and others. Although many of them provide graphical modeling interfaces and flexible functions, they do not appeal to a construction engineer as they are too complex and their operational framework is foreign to construction. In addressing this issue various researchers have attempted to enhance the CYCLONE method as it was easy to use but not flexible enough to handle all required complexities.

COOPS (Liu and Ioannou 1992) is an object-oriented, interactive simulation languages developed for simulating discrete construction systems. The provided interfaces allow the user to graphically construct a simulation model.

To model resource performance, especially allowing the mix of different size of resources at a QUEUE node, McCahill and Bernold (1993) presented the resource-oriented modeling concepts by attaching key attributes to each resource. A model library for a target user is also created.

Oloufa (1993) suggested the use of physical components of real construction systems as the classes or objects for Object-Oriented Simulation (OOS). Overlooking the complexity of such methods, this would be ideal for practitioners.

DISCO (Huang et. al. 1994) is an extension to MicroCYCLONE with a graphic interface which allows the user to graphically construct a CYCLONE simulation model, and includes a graphic monitoring function which enables the user to visualize the dynamic changes during the simulation process.

CIPROS (Tommelein and Odeh 1994) is an object-oriented and interactive simulation system which uses hierarchical concepts to model a construction process by matching resource properties with design and operations.

STROBOSCOPE (Martinez and Ioannou 1994) is a programming language for the simulation of construction processes.

HSM (Sawhney and AbouRizk 1995) combines the hierarchical concepts and the work breakdown structure concepts in constructing the simulation model for a construction project. The modeling process is divided into creating a work breakdown structure for a project (consists of operations and processes), developing process level simulation models, and specifying project required resources.

Modeling construction processes has always been hindered by the need of the modeler to be proficient in both simulation and construction operations. Major research in simplifying construction modeling includes graphical modeling interfaces (Liu and Ioannou 1992, and Huang et. al. 1994) and model reusability (Halpin et. al. 1990, and McCahill and Bernold 1993). A graphical modeling interface makes the process of building a simulation model easier since the user does not have to access lower level programming language details. The user still has to be proficient with simulation theory and the selected simulation language. Model reusability has been studied for construction. Halpin et. al. (1990) developed a standard library of simulation models that encompass a number of widely used construction processes. In a similar way, McCahill and Bernold (1993) implemented a library for a specific user (The U.S. Navy Civil Engineering Laboratory). While such libraries are effective, their major drawback lies is the fact that they must account for all of the possible user needs in terms of simulation models, a formidable task in general. With the diversity in construction practices and the uniqueness of construction projects such libraries though useful for targeted users, are not effective in general practice.

Construction simulation has been successful in academic research with limited successful applications in the industry. The major obstacles to its use by the industry are the complexities involved in constructing a model and the resultant time requirement -- the technique is not yet cost effective (Shi and AbouRizk 1994).

2.3.3 Simulation Optimization

One of the direct benefits of simulation is enabling the user to improve the performance of a construction system, which is called simulation optimization (Pritsker 1986). Simulation optimization methods that have been studied can be classified into four basic categories as follows (Azadivar 1992): Gradient-based Search Methods (GBSM) which derive from traditional non-linear programming techniques, are based on estimating the gradients of objective functions for approaching optimization (Pegden and Gately 1977). Stochastic Approximation Methods (SAM) involve recursive procedures for optimizing the theoretical regression function of a stochastic response surface (Robbins and Monro 1951). Response Surface Methods (RSM) are based on selecting points in the decision variable space, estimating these variables, fitting a series of regression models and attempting to optimize the fitted models (Biles 1974, Smith 1976). Detailed review of these procedures can be found in Glynn (1986) and Meketon (1987).

The parametric optimization methodologies described above can only be applied to situations when all variables are continuous, and thus fall short of covering many engineering applications which involve discrete parameters. Optimization for discrete event simulation has received little attention (Andradottir 1992, Goldsman, Nelson and Schmeiser 1991, and Yan and Mukai 1992).

In construction applications, simulation optimization mainly addresses the feasible resource allocation for a construction system from a practical point of view. In the context of this thesis, simulation optimization is defined as the process of searching a feasible resource allocation to maximize the production or to minimize the production cost of a construction system. Riggs (1979) developed a CYCLONE-based system with automatic sensitivity analysis capability. Touran (1987) reported a sensitivity analysis study to analyze the impact of each major variable on the tunnel advance rate. Berrios and Halpin (1988) performed optimization analysis for concrete construction by simulating and comparing different resource configurations. Halpin et. al (1989) presented a sensitivity analysis approach to resource analysis. The system interfaces with MicroCYCLONE and provides the user with results of multiple runs by specifying resource limits and other factors.

CHAPTER 3

RESOURCE-BASED MODELING METHODOLOGY

3.1 INTRODUCTION

Construction simulation is presently limited to academic research. The major obstacles to its use by the industry are the complexities involved in constructing a model and the resulting time requirements -- the technique is not yet cost effective (Shi and AbouRizk 1994). Simplifying existing simulation tools has been identified as one of the main issues in the research and development of construction simulation by the Simulation Panel of the Computerized Construction Research Workshop (Ibbs 1987). Graphic user interfaces and model reusability have been studied for simplifying modeling process in construction simulation. However, the user is still required to be proficient in both simulation and construction operations in building simulation models. This Chapter presents an automated modeling methodology -- resource-based modeling (RBM). It reduces the requirements for the level of knowledge in simulation, and is able to significantly simplify the modeling process in construction simulation.

3.2 CONCEPTS OF RBM

Hierarchical and modular modeling concepts were first presented by Zeigler (1976, 1984, 1985, and 1987). They have been useful in simplifying the construction process of simulation models, particularly for large and complex systems (Luna 1992). The basic components of the concepts include the 'atomic model', the 'model base or library', and 'coupling'. An atomic model is a basic and unique description of a particular process. A model library consists of numerous atomic models, which are to be used in various combinations to construct a high level model. Coupling is the act of combining related atomic models. Figure 3-1 illustrates the process of coupling atomic models A and B in the construction of the higher level model AB.

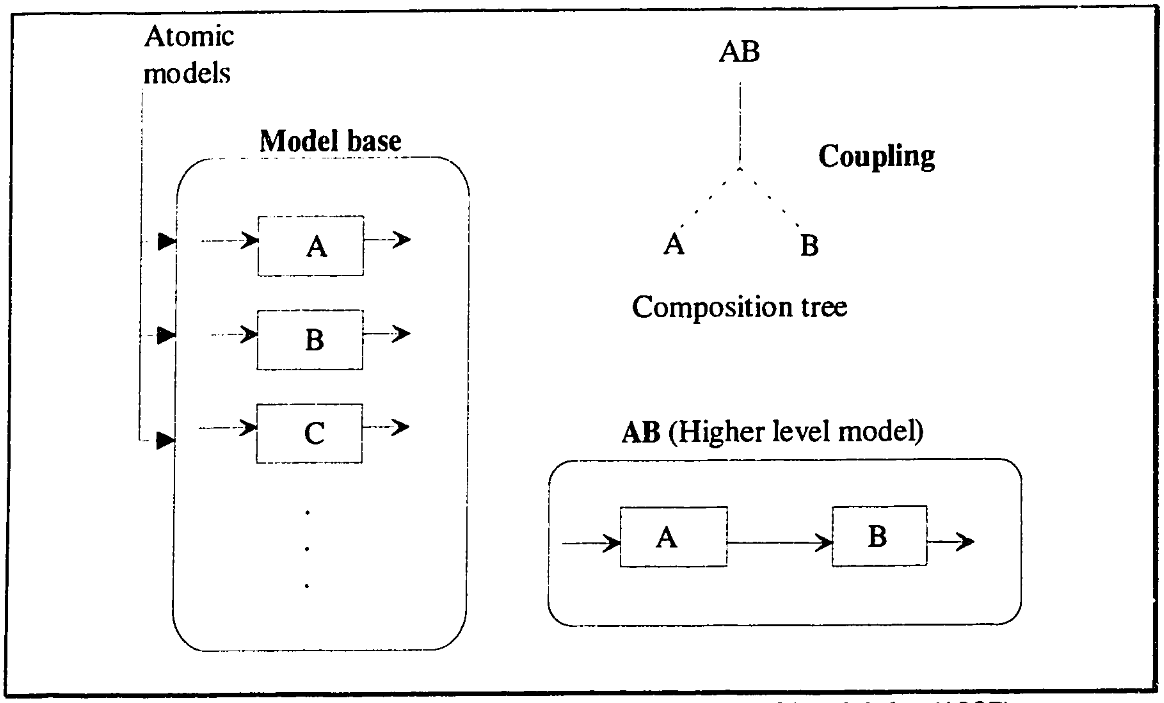


Figure 3- 1 Modular modeling concepts presented by Zeigler (1987)

In general terms, the following tasks and issues must be addressed to apply the hierarchical approach to modeling construction projects:

1. The atomic models that are to be included in the model library must be defined and designed.

2. Coupling procedures that address the actual requirements of construction projects must be developed. Zeigler (1987) suggested creating new models by combining two or more atomic models through model input and output ports. The authors' early

experimentation with this approach showed that two or more atomic models cannot always be directly linked through simple input/output ports. Various types of linking structures must be defined and implemented to facilitate the coupling process.

3. A means of integrating the attributes and boundaries of the physical system environment (e.g. project site information) must be incorporated into the modeling process.

Many real-world systems are characterized by dynamic resource interactions (e.g. different types of equipment in earthmoving construction). Although each construction system is unique, the operating processes of its component resources are usually somewhat generic. They can be pre-defined as atomic models and stored in a model library. Atomic models from the library can then be incorporated with project-associated data to form project-specific atomic models. The environment will then identify appropriate linking structures and assemble the working model for a given project.

As noted in Chapter 2, process libraries of construction simulation for reusability purposes were studied (Halpin et. al. 1990, McCahill and Bernold 1993). A construction process is usually unique, and is related to selected resources, site conditions, and management. The major drawback of a process library is its limitation in flexibility for general uses.

The atomic model library defined in an RBM focuses on the low level process of a specific resource as detailed in Section 3.5. An atomic model can produce multiple submodels depending on resource and project specifications provided by the user. For example, there is only one atomic model for the loading operation of excavators. As many "loading" submodels as required for a project can be generated from the same atomic model by incorporating actual project data and user-specified excavators. Through defined linking structures, all submodels can be assembled into one working model. Compared with traditional process model libraries, the RBM is more flexible in handling different situations of various construction projects. For instance, there are two earthmoving projects: project 1 has two loaders and one hauling fleet; project 2 has two loaders and two hauling fleets. A process model library must have two separate process models to accommodate the two cases. The RBM can model both situations. Two loading submodels and one hauling submodel are generated and assembled through a linking structure for project 1. Two loading submodels and two hauling submodels are generated and assembled through another linking structure for project 2.

3.3 OVERALL ENVIRONMENT STRUCTURE

Conventionally a user must understand both simulation theory and the selected simulation language to construct a simulation model. The RBM is a modeling framework which acts like a pre-processor for a simulation language. Through this pre-processor the user can construct a simulation model by simply specifying required resources and project-related information; the user does not have to be proficient with simulation. Eight basic components form the core of an RBM environment:

1. a database to store resource attributes;

2. an atomic model library which includes all types of resources for a specific type of construction project;

3. a user interface that allows the user to specify required resources, project-related resource attributes, and other project information;

4. a module which can convert physical site conditions to formats acceptable to the simulation model. An example of this conversion is the computation of the duration of a work task from given physical project site conditions;

5. an atomic model generation module which can combine resource attributes and project-related information with atomic models in the library to produce project-specific atomic models;

6. a knowledge-based module which can identify and generate proper linking structures to suit the atomic models and project;

7. a module which can assemble all atomic models through linking structures to generate a working simulation model;

8. an interface which can call the selected simulation language and allow the user to experiment with the generated model.

Figure 3-2 illustrates the structure of the RBM environment and the interactions between the eight basic components. The following sections of this Chapter describe the individual components in greater detail.

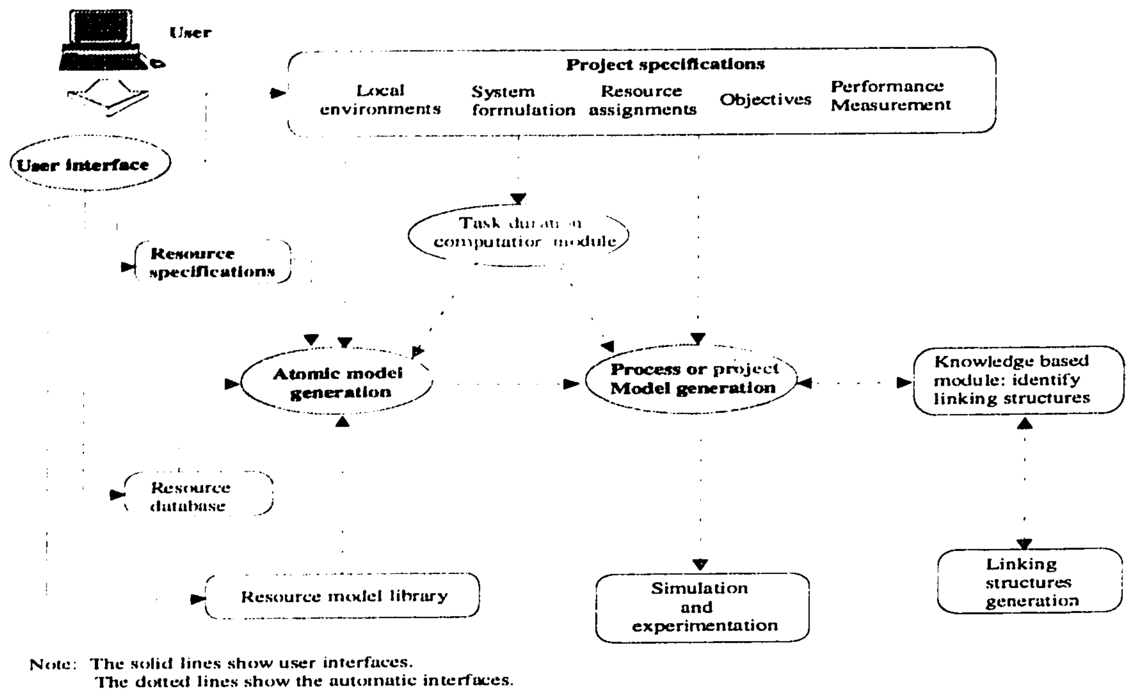


Figure 3- 2 Overall system structure

3.4 RESOURCE ATTRIBUTES

The "attributes" component of an atomic model is a vector which describes the key features of a resource. The format can be:

ATTR(common attributes, project-specific attributes) (2-2)

The common attributes of a resource describe its properties which are identical from project to project. Those attribute values are established in a resource database system which is maintained by the company. This database can be modified as required to suit a company's changing status. The sample common attributes of a resource are: available number, rented or owned, working reliability, and operating cost. More other common attributes can be added to the database as necessary in the implementation.

The project-specific attributes of a resource describe its properties while it is assigned to a specific project (e.g. project required number, supporting labor, management level, etc.). Those attributes will require the user to specify values according to a specific project situation.

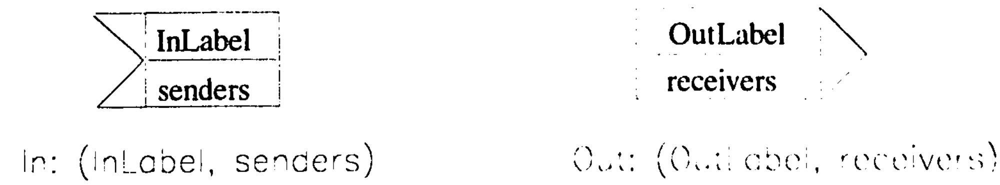
3.5 ATOMIC MODELS AND ATOMIC MODEL LIBRARY

Resources can be classified as either active or passive. An active resource, as the name implies, actively performs an operation. Equipment and labor are active resources, while material is passive. An active resource is always associated with a process when it is used for a specific operation. For example, the loading operation of excavators always has four basic work tasks: load bucket, swing loaded, dump bucket, and swing empty, although the durations of the four work tasks may vary for different types of excavators and project site conditions. The loading atomic model of excavators, therefore, has four basic work tasks. The task durations are given by the user through user interface according to the selected excavator and project site conditions.

In simulation, construction processes are modeled in two formats -- graphic and textual. The graphic format uses defined symbols to develop schematic representations of a construction process. The textual format represents the same process model in a code syntax. In a specific simulation environment, the two formats are interchangeable. Normally, when the user creates a graphic model, a corresponding textual model is automatically created at the same time. However, the two formats have different functions. The graphic format allows the user to visualize the operating process of a resource and its communication ports. The graphical representation has proven to be very important for simplifying construction modeling (AbouRizk 1994). The graphic format, however, cannot be directly used as an input file to a simulation language. Instead, it must be converted to a code (textual) format in order to experiment with the created model using a simulation engine.

CYCLONE-based construction simulation methodology was invented by Halpin in 1973 and has since been enhanced by many researchers. Because of its simplicity, CYCLONE is one of the most common simulation languages used in construction simulation. Therefore, the graphic format of the atomic model representation in the RBM uses CYCLONE. CYCLONE's limited modeling and simulation function does not allow the user to model and simulate complex and large construction systems. However, in the RBM, the user will not have to directly deal with the textual representation of a simulation model; therefore, an efficient but complex textual format can be selected without elevating the required level of understanding. The main criterion for selection of the textual format should be of flexibility so that the user can generate and experiment either simple or complex models in accordance with practical requirements. For the trial implementation of RBM for earthmoving construction (detailed in Chapter 4), SLAM II textual format was selected to represent the atomic models in the library.

Communication ports identify how models can interface with each other. There are two types -- input and output ports. Input ports receive messages from other models. Output ports send messages to other models. The two ports are illustrated graphically and textually in Figure 3-3.



(a) Input port

(b) Output port

Figure 3- 3 Communication ports

"InLabel" and "OutLabel" represent the names associated with the ports. The "senders" and "receivers" are vectors representing the possible sources and targets of messages. The "senders" of an input port must be output port(s) in other models. Similarly, 'receivers' of an output port must be input port(s) in other models. Models can be coupled only when their communication ports match each other. If an input port cannot find corresponding 'senders' or if an output port can not find corresponding 'receivers', the input port or output port will be disabled and can be removed from the original model. An input port can have more than one 'sender'; and an output port can have more than one 'receiver'. For example, one fleet of trucks may be assigned to work with two backhoes allocated to the same location.

Different types of construction projects require different kinds of resources. For instance, earthmoving projects require tractors, trucks, excavators, loaders, and other earthwork equipment, while pipeline construction requires a different set of equipment including trenchers, excavators, pipelayers, and welding rigs. Therefore, a separate model library should be specifically designed for each type of construction project. Libraries could be assembled according to the construction classification proposed by Halpin and Woodhead (1980) shown in Figure 3-4.

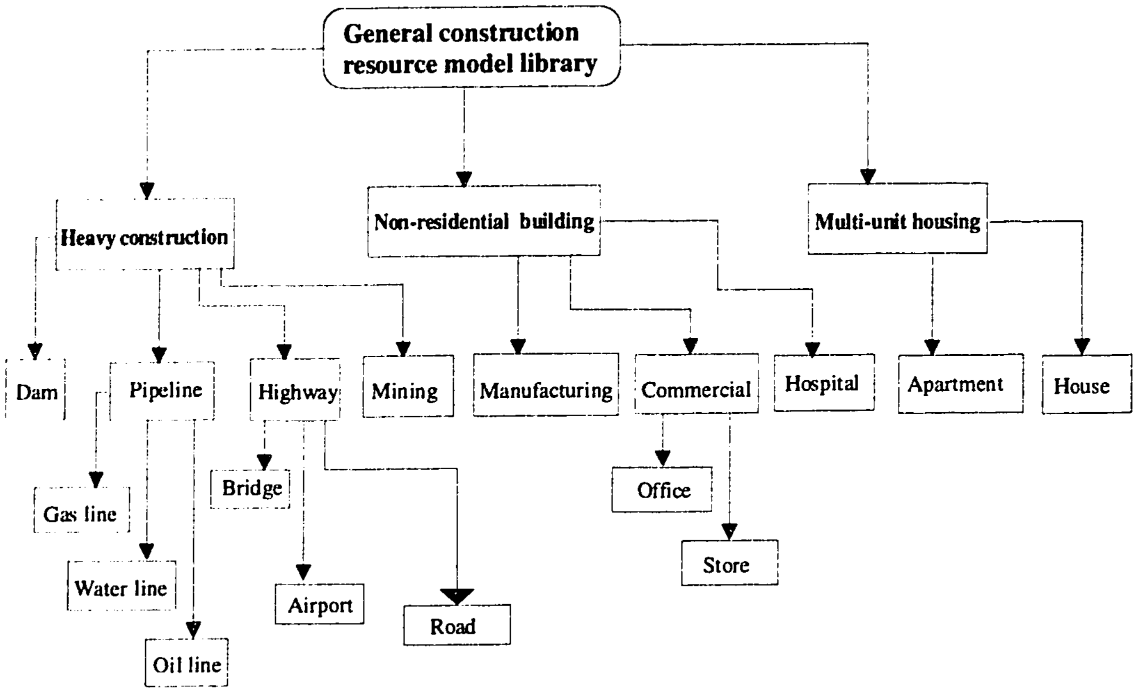


Figure 3- 4 Model library breakdown structure

3.6 RESOURCE SPECIFICATIONS

Resource specification accommodates a user interface which allows the user to specify the required resource combination and project-specific resource information for a project. The specification process is simply picking up required resources from the embedded resource database, and attaching with project-specific attributes (e.g. required number).

3.7 BOUNDARY AND ENVIRONMENT SPECIFICATION

The boundary and the environment of a construction system identify the fundamental features of its operating conditions and will affect the entire system's performance. They must, therefore, be incorporated into the simulation model for the true representation of a real system. For example, the simulation results of a pipeline construction project in the cold weather cannot be applied to a similar project in the hot weather (assuming environment being the only difference.) Boundary and environment conditions are general terms for all systems. In the context of this thesis, "project" is used to refer to the boundary and environment of a construction system.

Project specification requires detailing of the physical features which are part of the overall simulation model. All major factors that affect the real construction processes should be specified through the project specification user interface. Five aspects of physical features have been identified and should be specified by the user. They are: 1) system specification, 2) resource assignment, 3) local environment, 4) measurement, and 5) objectives.

3.7.1 System specification

Halpin and Riggs (1992) presented a hierarchical representation for construction projects based on the operational considerations shown in Figure 3-5. A construction project or operation can be defined in terms of processes, which are collections of work tasks. A work task is a readily identifiable component of a construction process or operation. The various work tasks are logically related in accordance with the technology of the construction process and the work plan to form a process.

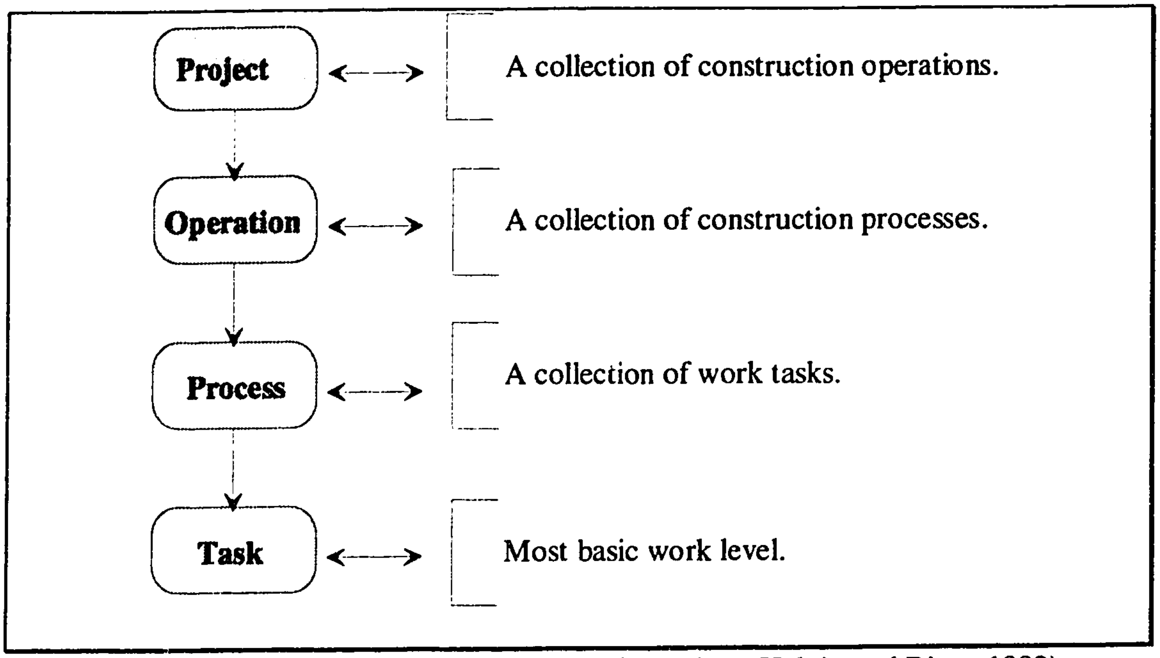


Figure 3- 5 Hierarchical representation of a project (Halpin and Riggs 1992)

Resource atomic models in the RBM focus on the "process-task" portion of this hierarchy. For developmental purposes, we can define a new level between "process" and "task" termed the "resource-based process" (*r-process*), which describes the operating process of a resource through a series of tasks. For instance, while an excavator is used for loading purposes in an earthmoving project, its *r-process* includes four basic work tasks: load bucket, swing loaded, dump bucket, and swing empty. An *r-process* has a corresponding atomic model in the library as described in previous sections. It is the basic component that defines a construction project, an operation, or a process.

In system specification, the user is required to represent a project in basic *r-processes*, and to define logical interrelationships among these *r-processes*. In other words, system specification defines the makeup of a system in terms of its constituent *r-processes*. For example, suppose a construction system has four basic *r-processes* (labeled as *r-processes* 1, 2, 3, and 4). *R-process* 1 is followed by *r-process*es 2 and 3, which are followed by *r-process* 4. A box can be used to represent an *r-process*, and an arrow can be used to model the logical sequence between two *r-process*es as shown in Figure 3-6.

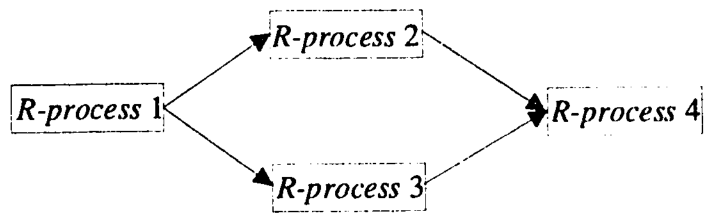


Figure 3- 6 An example *r-process* specification

Each type of construction project can be defined in terms of its own *r-processes*. For instance, earthmoving projects can include: bulldozing, scraping, loading, hauling, transporting, placing, and compacting. Pipeline construction can have: right-of-way, stringing, trenching, bending, welding, lowering-in, backfilling, and testing. It is necessary to define descriptive *r-processes* for each type of construction project.

3.7.2 Resource assignment

An *r-process* is always performed by a resource. For example, an excavator is used for the loading *r-process* in earthmoving. Different resources may perform the same *r-process* with different operating processes. For instance, the basic work tasks for a loading *r-process* with an excavator are load bucket, swing loaded, dump bucket, and swing empty; for the same *r-process* with a loader the basic work tasks are load bucket, move loaded, dump bucket, and return empty. Additionally, one resource can be used for different operations so it may be represented by more than one *r-process*. For example, the model of a tractor used to push dirt is different from that used to assist a scraper.

After a constructio project or process has been defined in terms of *r-processes*, each resource should be assigned to corresponding *r-processes* to achieve planned operation. The following syntax is used to assign resources to *r-processes*.

Assign: *Resource i* To *R-process j* (3-3)

3.7.3 Local environment

The local environment is defined as the site conditions associated with each *r-process*. Because *r-processes* are associated with resources, local site conditions influence the operation of resources. The local environment specification is project-type, *r-process*-type, and resource-type related. For example, loading pattern and soil type affect a loading *r-process*; distance and road condition affect the hauling *r-process*. The basic *r-processes* must be broken down into groups, with each group having a set of factors to be specified by the user, as illustrated in Figure 3-7 for earthmoving construction.

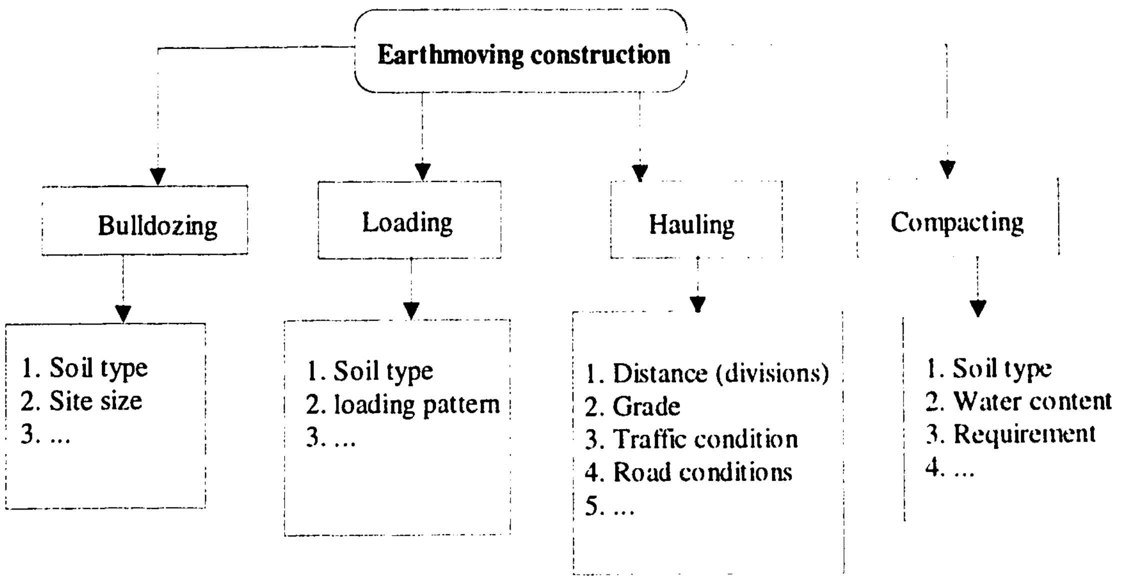


Figure 3- 7 Local environment specification for earthmoving

3.7.4 Measurement

The measurement of an *r-process* is defined as the production per cycle of a resource while performing its assigned duty. When a resource is used for different *r-processes*, the measurement of its operation may be different for each *r-process*. For example, when a backhoe is used to load dirt into a truck, measurement of the *r-process* may be cubic meters, when the same backhoe is used to dig a trench, the measurement would be lineal meters.

3.7.5 Objectives

The objective of an *r-process* defines the termination of its operation. For example, if a particular *r-process* measurement is 10 cubic meters per cycle and the *r-processes* objective is specified as 125 cubic meters, the operation of this *r-process* will terminate after 13 cycles. The entire system's objective follows the same principle.

3.8 TASK DURATION COMPUTATION

There are two alternative ways to specify local environment: 1) the user converts all physical site conditions into corresponding time parameters or correction factors, or 2) the user directly specifies the real physical site conditions. If 2) is chosen, corresponding time parameters or correction factors must be calculated for the environment. For example, the hauling time of a truck can be calculated from rolling resistance, grade, traffic conditions, and hauling distance. For a complex construction system, a knowledge-based system may be necessary to calculate time parameters from physical site conditions.

3.9 PROJECT-SPECIFIC ATOMIC MODEL GENERATION

Atomic models in the library only describe the basic logic structures of corresponding resource operations. No actual data are included in these models. For example, there is only one loading atomic model in the library for all excavators. The operating parameters for each excavator may be different from project to project with varying site conditions. The atomic model generation module constructs project-specific atomic models from library-resident atomic models for each specified *r-process* in accordance with specified resource attributes, site conditions and *r-process* objectives. These generated models become the bases for assembling the entire model for a project.

3.10 ENTIRE MODEL GENERATION

An entire simulation model is the final result of the modeling process. It is obtained by combining generated *r-process* models through linking structures. The concurrent generation of supporting text files may also be required depending on the selected simulation language. SLAM II, for instance, requires a control file, a scenario file, and/or a user insert file to form a complete simulation-ready package. The generation of these supporting files is detailed in Chapter 4. This section addresses the linking structures in a general manner.

Project-specific models are like building blocks. A required building can be generated by properly linking the appropriate blocks. Linking structures in the RBM are used to link involved models into one entire simulation model. A linking structure is a submodel which receives input from one *r-process* model and transfers it to another *r-process* model. The purpose of a linking structure is to correctly assemble related *r-process* models according to their characteristics.

The simulation entity is defined as information flowing through a simulation model (Pritsker, 1986). An entity is any object, resource, unit of information, or combination thereof which can define or can alter the state of a simulation system. For example, in an earthmoving problem, the status of the system could be represented by the number of busy loaders and the number of trucks waiting for loading. Simulation entities traverse through a simulation model as the system's status changes. They can also traverse from one atomic model to another through communication ports. Normally, simulation entities in different submodels have different meanings and associated measurements. In some cases, the entities in two models are dimensionally equivalent. In other words, one released entity from a model will be directly routed to its following model without any change. For instance, after foundation process releases an entity, substructural process can start. A "direct link" can be defined to model this situation to directly route a simulation entity from a model to its following model(s). However, the communication between models cannot always be directly accomplished because the entities in these models are not dimensionally compatible. For instance, where a backhoe excavator can load 8 m^3 soil into a truck in each bucket, and the truck can hold 50 m^3 soil, six entities in the loading model equal to one entity in the haul model. At this situation, a transition should be designed in the linking structure to balance the difference between the entities in different models. This type of linking structure is defined as "indirect link".

In this section, an ellipse surrounding a word is used to represent a generic function node in the graphical representation of those various linking structures. It takes specific forms in a selected simulation language.

3.10.1 Direct linking structures

Direct linking structures will not alter simulation entities during the transfer process. The output of a model may be required as input to one or more following models, and the outputs of several models may be required as inputs to a single following model. These various scenarios can be detailed as follows:

1. One-one link

A one-one link is the simplest scenario. The output of one model is required as the sole input to another model. Using arrows to define the coupling process, only one arrow is required to link the two models, as shown in Figure 3-8. For example, a welding crew can start welding after pipes are positioned, and this one-one link can be used to couple the "pipe positioning" and "welding" models.

Figure 3- 8 One-one direct link

2. One-multiple link with all branch releases

Where multiple models require the single output of a preceding model, a "*continue*" function node, which releases simulation entities to each of the following models, can be added to the linking structure as shown in Figure 3-9. At the situation while multiple activities can start after the foundation is completed, this one-multiple link can be used to couple them.

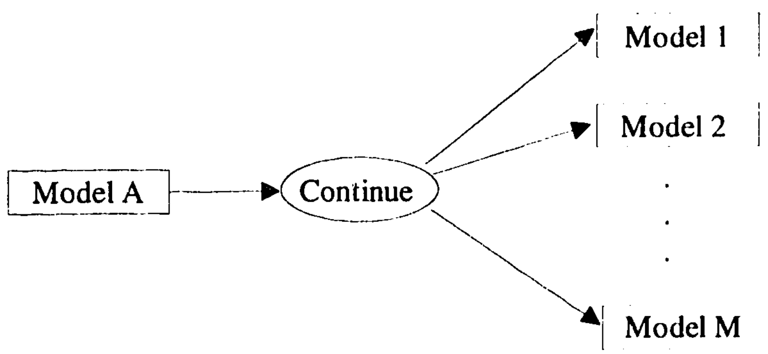


Figure 3- 9 One-multiple direct link

3. One-multiple link with one branch release

Where multiple models follow a single model, and only one of them can be released at a time, a function "*select*" node can be added to the linking structure as shown in Figure 3-10. The selection rule could be "cyclic" or "priority" depending on the selected simulation language. This link can be used for the situation where an empty truck has to select one from the multiple source loading areas.

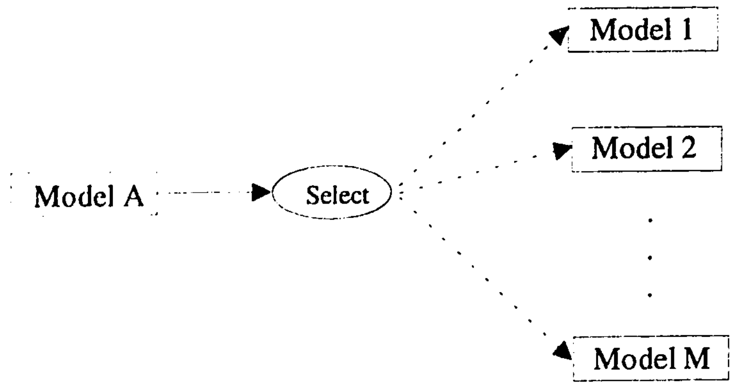


Figure 3- 10 One-multiple direct link

4. Multiple-one with multiple releases

If multiple models are followed by a single model and all outputs are to be routed to the following model, the treatment is similar to the one-one situation. For each preceding model, an arrow routes its output to the following model. This structure is illustrated by Figure 3-11. If multiple loaders are used to serve one hauling fleet, this link can be used.

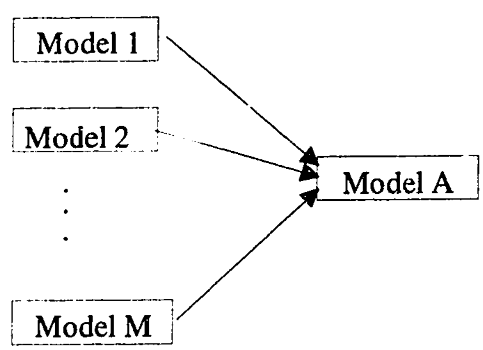


Figure 3- 11 Multiple-one direct link with multiple releases

5. Multiple-one link with one release

Where multiple models precede a single model, and one output is required from each of them to release the following model, a function "*consolidate*" node can be added to the structure as shown in Figure 3-12. This link can be used for the situation where pouring concrete cannot start until both formwork has been completed and concrete has been delivered.

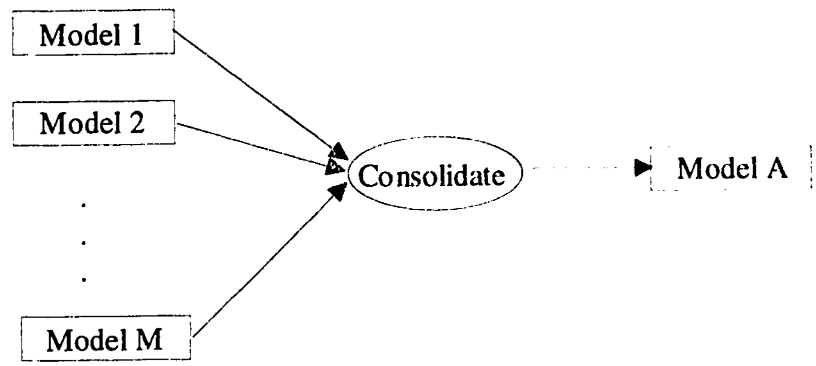


Figure 3- 12 Multiple-one direct link with one release

3.10.2 Indirect linking structures

Indirect linking structures are used to couple models in which simulation entities are not dimensionally compatible. Assuming the simulation entity in model A is measured in units X, and the simulation entity in model B is measured in terms of units Y. The transition structure can divide one entity of model A into X units that are sent to a queue node, then Y entities released from the queue node are combined up into one entity before it flows into model B. Various indirect links can be defined as follows.

1. One-one indirect link

Similar to a direct one-one link, an indirect one-one link implies a single model is followed by another single model. In this case, the entity released from model A is divided into multiple units which are routed to a queue node. Then multiple entities released from this queue node are combined into one entity which is routed to Model B as shown in Figure 3-13. This link is used to couple a "bulldozing" model and a "loading" model because the production achieved in one operating cycle of a bulldozer is normally not equal to the production achieved in one operating cycle of a loader.

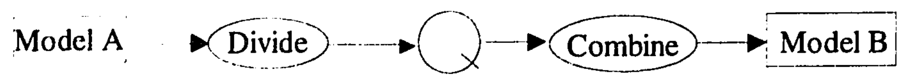


Figure 3- 13 One-one indirect link

2. One-multiple indirect link

If multiple models require the output from a single preceding model, the first part of the linking structure is identical to the one-one scenario. After a queue node, each following model has a separate "combine" node as shown in Figure 3-14. This link can be used for the situation where one bulldozing process is serving multiple loading processes,.

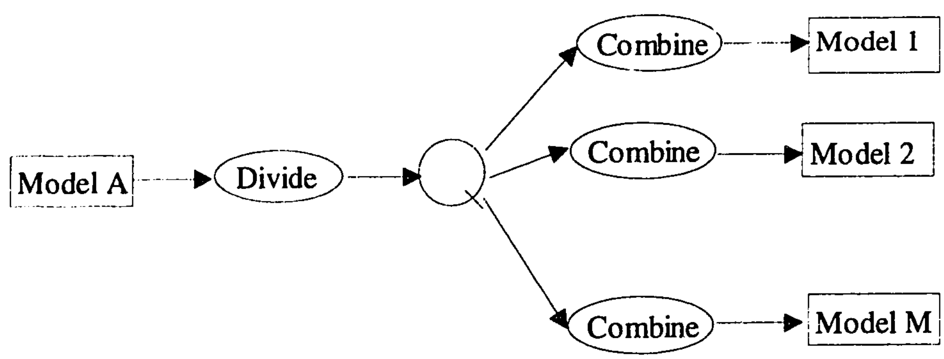


Figure 3- 14 One-multiple indirect link

3. Multiple-one indirect link

If multiple models are followed by a single model, all entities released from preceding models are to be divided and routed to a queue node. Then entities released from the queue node are combined and routed to model B. This structure can be shown as in Figure 3-15. This link can be used for the situation where multiple hauling fleets are followed by one spreader.

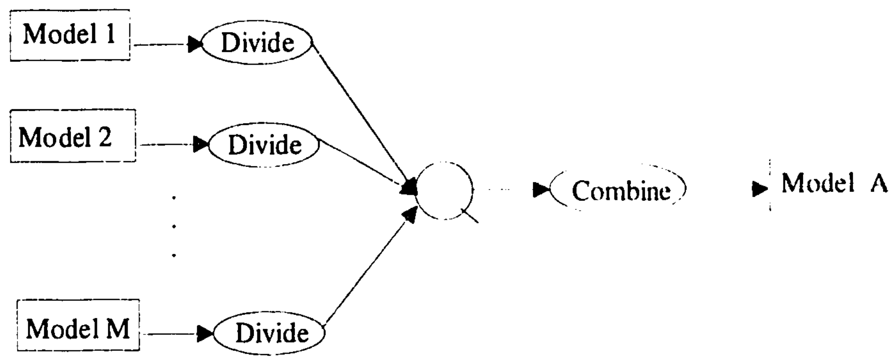


Figure 3- 15 Multiple-one indirect link

3.10.3 Identify required linking structures

All project-specific *r-process* models should be linked together to form a complete simulation model for a given construction process or project while following the logical sequences defined by the modeler in the project specification. The linking structures required to couple related *r-process* models depend upon the properties of involved models and the number of models preceded and followed. The properties of involved models determine whether direct or indirect links should be used. A hauling *r-process* model, for instance, can only be coupled with a spreading *r-process* model through an indirect link in an earthmoving RBM environment because the simulation entities are not dimensionally compatible in both models.

The specific linking scenario required in either direct or indirect linking category depends on the number of models preceded and followed. If a model is preceded by three models, for instance, then a "three-one link" should be used to couple this model with its preceding models; and if it is followed by two models, a "one-two" scenario should be used to couple it with its following models. The numbers of models preceded and followed can be easily identified according to the number of *r-processes* and logical sequences specified by the modeler in project specification.

3.10.4 Embedded flexibility of an RBM environment

A traditional process model library can only provide a limited number of process models which have already been assigned to it. If a library does not have a process model corresponding to a construction process, it fails to provide a simulation model for that situation. A process model library cannot likely accommodate all scenarios required for modeling different projects because of the uniqueness of construction projects. Therefore, a process model library is not likely to be flexible enough to satisfy the requirement for modeling general construction processes or projects.

The flexibility of an RBM derives from its atomic models which can produce multiple project-specific *r-process* models by incorporating resource and project information into the atomic models. The same type of multiple project-specific *r-process* models can be produced as illustrated in Figure 3-16.

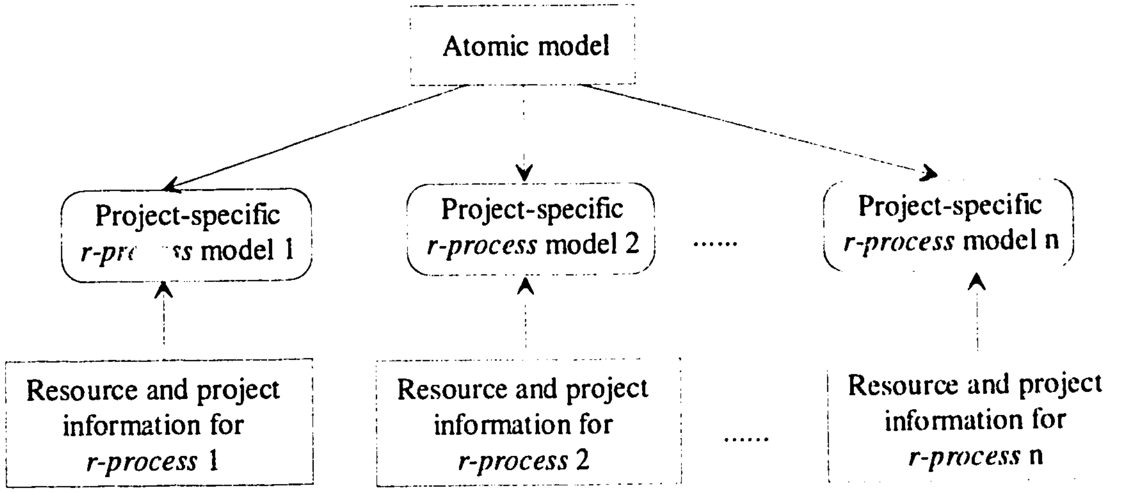


Figure 3- 16 Generating process of project-specific r-process models

For example, three project-specific loading *r-process* models will be generated from the same loading atomic model if there are three loading *r-processes* in a project. The modeler can specify specific resource and site conditions for each *r-process*. For instance, the modeler can assign a Hitachi-EX1800 backhoe for loading *r-process* 1, a Hitachi-EX1100 for loading *r-process* 2, and a CAT-992 loader for loading *r-process* 3. The work task durations for each of the three *r-processes* can be separately obtained from its assigned resource and physical site conditions provided by the user in project specification. Three identical loading *r-processes* can then be generated by incorporating all specified information as illustrated in Figure 3-17.

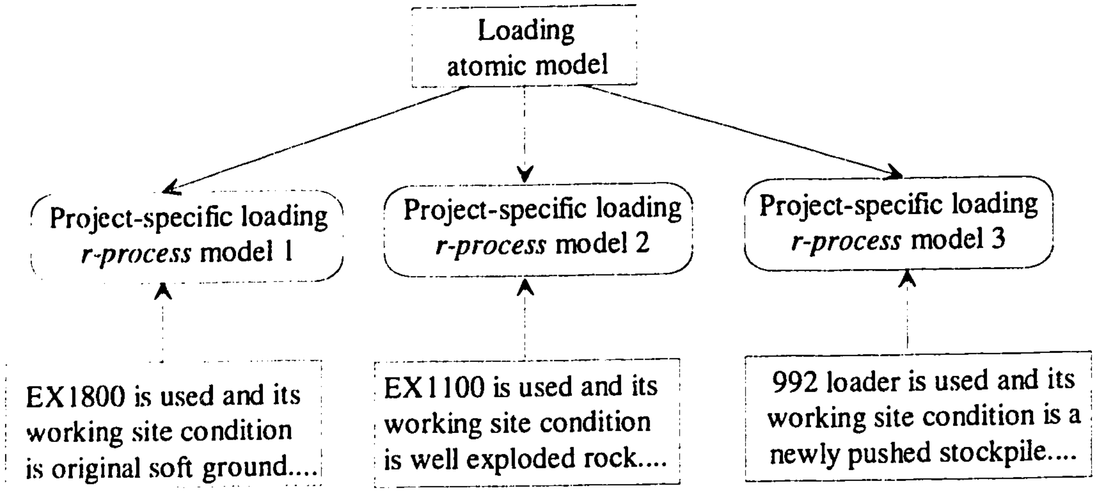


Figure 3- 17 A sample generation process of three loading *r-process* models

The flexibility of an RBM lies in its ability to derive the required relationships among all *r-process* models as defined by the modeler in the project specification according to the requirements of a real world construction project. All *r-process* models are then assembled together through identified linking structures by following the specified logical sequences. Therefore a unique project or process model can be constructed for each given project. For the same example of the three loading *r-processes*, each of them can have its own preceding and following *r-process* models based on project specification, e.g.: loading *r-processes* 1 and 2 are not preceded by other *r-process*, and each is followed by a separate hauling *r-process*; Loading *r-process* 3 is preceded by a soil preparation *r-process* and followed by a spreading *r-process*. The modeler can use the same procedure for other *r-processes*. A unique simulation model can then be generated.

In conclusion, the flexibility of an RBM is not limited by the number of *r-processes* or the logical relationships among them. A unique simulation model can be generated as needed according to the user specified resource and project information for a project.

3.11 SIMULATION AND EXPERIMENTATION

After a simulation model is constructed and has been validated, the user should do some strategic and tactical planning to establish the experimental conditions for the simulation runs. This phase of simulation involves the exercising of the simulation model and the interpretation of the outputs. In construction simulation, the user usually wants to know the maximum production rate, the total construction duration, and the optimum resource allocation for a construction project. Multiple simulation scenarios are always required to achieve these objectives. In Chapters 5 and 6, two optimization techniques are presented to assist the user to locate optimum resource allocation for construction simulation.

As noted previously, this research provides a pre-processor for a simulation engine in assisting modeling. A simulation language is used to for the simulation and experimentation for a constructed model.

3.12 A MODELING EXAMPLE

To illustrate the RBM concept, consider an earthmoving operation, with five *r-processes*: soil preparation, loading at location 1, loading at location 2, hauling, and spreading. The sequence of the five resource processes are specified as shown in Figure 3-18. Five resources have been specified: a CAT-D8 dozer, a UH-501 backhoe excavator, CAT-777 trucks, and a CAT-7 dozer. The *resource assignments* are: CAT-D8 for soil preparation, UH-501 for loading at location 1 ; EX-300 for loading at location 2, CAT-777 trucks for hauling, and CAT-D7 for spreading. *Local environments* are: hard clay grou ·. for CAT-D8 (hard pushing), loose stockpiled clay (easy cut) for UH-501 and EX-300; a haul route consisting of a section of 10% inclining grade and a section of busy traffic for the trucks, and loose stockpiled clay (easy pushing) for CAT-D7. *The system objective* is to spread 100,000 m^3 of excavated soil at the dump area.

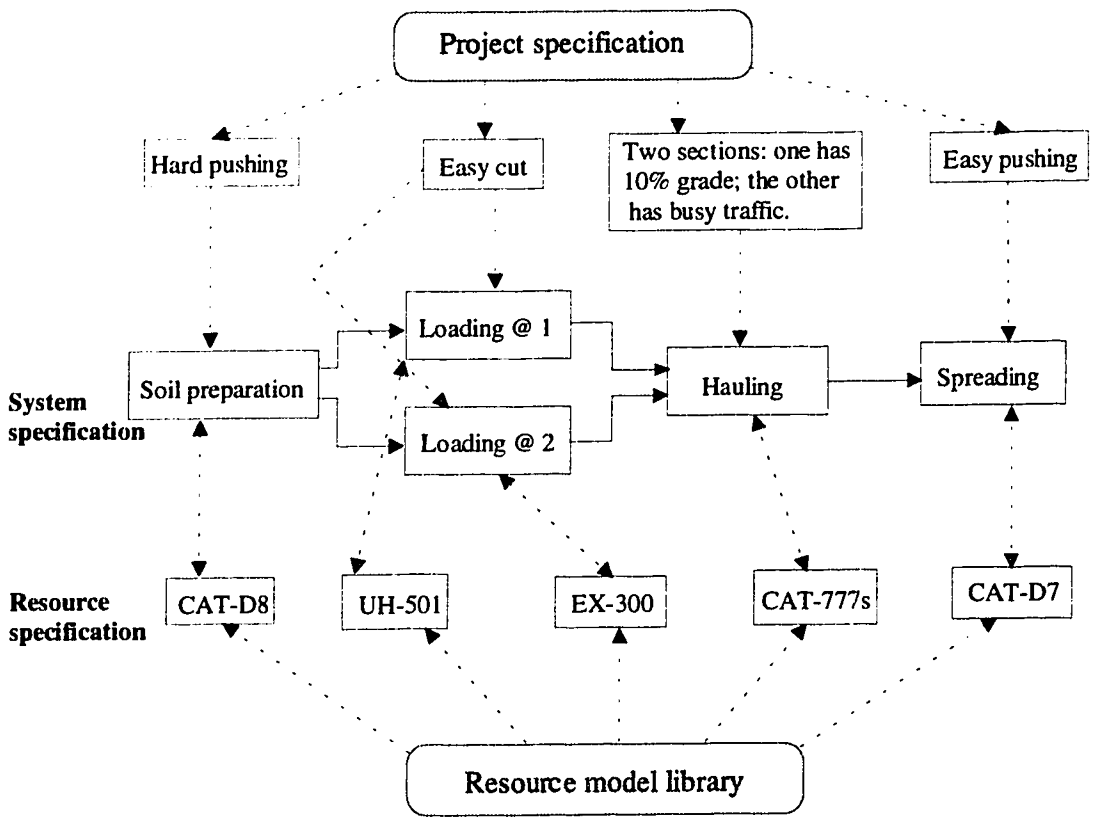


Figure 3- 18 An earthmoving modeling example

All information is specified by the user through provided interfaces, and will be incorporated into corresponding atomic models to generate project-specific *r-process* models. One soil preparation *r-process*, one hauling *r-process*, one spreading *r-process*, and two loading *r-process* models are to be generated for this sample project. The linking structures are then automatically identified as required to assemble the five *r-process* models: one-multiple indirect link between soil preparation and loading, multiple-one indirect link between loading and hauling, and one-one indirect link between hauling and spreading. The entire project simulation model is constructed by assembling these *r-process* models by following the logic sequences provided in the system specification.

After the simulation model is constructed, the user can switch to a simulation environment to experiment with the model.

3.13 CONCLUSIONS

The RBM described in this Chapter utilizes generic programming tools which, with input provided from a user specification interface, automatically construct a simulation model for a given project. The user needs to be familiar with the construction operations, project site conditions, and resources that are used in the project, but no simulation language expertise is required in this modeling process. This approach may significantly reduce the time requirement in the modeling process of construction simulation, improve the cost-effectiveness of construction simulation, and allow the transition of construction simulation from research institutions to the industry. In Chapter 4, RBM will be specifically implemented for modeling earthmoving operations.

CHAPTER 4

AN AUTOMATED MODELING ENVIRONMENT FOR EARTHMOVING OPERATIONS

4.1 INTRODUCTION

Earthmoving is a common construction operation typical to building foundation work, dam construction, airport construction, road construction, strip-mining, and other work. Peurifoy and Ledbetter (1985) described earthmoving operations in four basic processes including excavating, hauling, placing, and compacting. For this equipment-intensive construction system, mainly deterministic methods are currently used for selecting and matching different types of equipment for a project.

Earthmoving equipment can be categorized according to its performed functions. Nunnally (1980) groups earthmoving equipment into four classes: 1) excavating and lifting, 2) loading and hauling, 3) compacting and finishing, and 4) rock excavating. Commonly used earthmoving equipment includes tractors, excavators, scrapers, loaders, trucks, compactors, and graders. Tractors can be further divided into crawler or wheel mounted. The basic function of a tractor is to push or pull loads. With a bulldozer blade attached, its function can be expanded to include the cut/stockpile and spreading operations. The excavating family includes shovels, backhoes, draglines and clamshells, trenching machines, and others. Scrapers are combined loading and hauling machines. They can load, haul and discharge materials without assistance, although it is common to use push tractor to provide assistance in hard-cut areas. Spreading is the process of placing earth in the fill zone. Compaction increases the density of soil by mechanically forcing the soil particles closer together, thereby expelling air from the void space in the soil. Grading is the process of bringing earthwork to the specified shape and elevation or grade.

Many techniques have been tried to improve earthmoving operations. Easa (1987) developed a linear programming model which can solve the earthwork allocation problem with non-constant unit costs. Christian and Caldera (1987) developed another operational research model which considers and addresses the swell and shrinkage of soil. Recently, more researchers are attempting to apply expert systems to assist in the selection of earthmoving equipment (Alkass and Harris 1988, Amirkhanian and Baker 1992). Vanegas et al (1993) present a CYCLONE-based simulation model for muck-hauling operations in tunnel excavation. A simulation model for strip mining construction has been studied by Shi and AbouRizk (1994). Simulation is an effective approach to study this type of construction system. However, we need to simplify it and to make it a useful tool for practitioners.

This Chapter illustrates the implementation of the RBM to general earthmoving operations. The atomic models, atomic model library, resource specifications, project specifications, and modeling process are discussed. The implementation medium is PC computers using VISUAL-BASIC programming language.

4.2 ATOMIC MODELS AND ATOMIC MODEL LIBRARY

The atomic model library contains atomic models for commonly used equipment categories as they would be used for specific operations. Graphical and textual representation are used to describe atomic models. For its simplici.y, the graphical representation uses CYCLONE methodology. The textual representation uses SLAM II syntax. The final generated simulation model can be experimented with SLAMSYSTEM.

Some equipment can perform only one specific operation (or *r-process* as defined in this thesis). Certain trucks, for instance, can haul materials only from one location to another. For this type of equipment, a unique atomic model is created and stored in the model library. However, some other machines are flexible and can be used for different operations. Tractors, for example, can be used to bulldoze, to assist scrapers, or to spread earth fill. For these types of equipment, multiple atomic models corresponding to all possible working operations should be created and stored in the model library. Further, different types of equipment can be used for a particular operation. They can share the same atomic model if they have the same operating process; otherwise a separate atomic model should be created for each resource. For example, both tractors and graders can be used to place earth fill and they can share the same placing atomic model. Both loaders and excavators can be used for loading with different operating processes, therefore, two loading atomic models for loaders and excavators respectively are necessary. In the remainder of this section, the atomic models included in the atomic model library will be described.

4.2.1 Tractors for bulldozing

When a tractor is bulldozing, five work tasks are involved: position the tractor (BD1), cut soil (BD2), move the soil to dump (BD3), dump the soil (BD4), and return for the next cycle (BD5). One output communication port follows the dump task. The graphic (CYCLONE) representation of the atomic model is illustrated in Figure 4-1. The SLAM II network statement is shown in Figure 4-2.

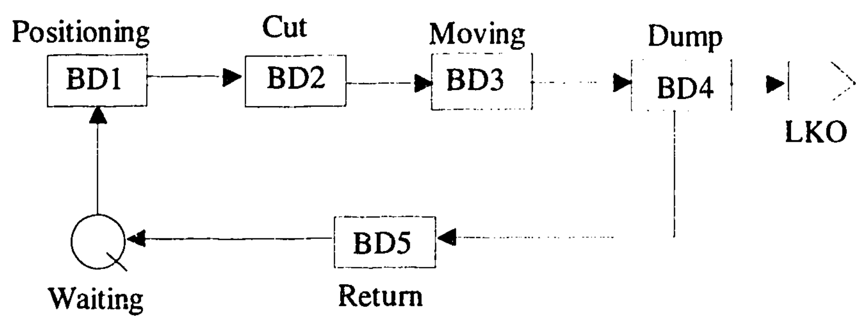


Figure 4- 1 Bulldozing atomic model

```
BA1  AWAIT (BN1),TRACTOR/1,1;
     ACTIVITY/BD1,BT1,,BG1; positioning;
BG1  GOON,1;
     ACTIVITY/BD2,BT2,,BG2;cut;
BG2  GOON,1;
     ACTIVITY/BD3,BT3,,BG3; move
BG3  GOON,1;
     ACTIVITY/BD4,BT4,,BG4; dump;
BG4  GOON,2;
     ACTIVITY,,, CNT;
LNK  ACTIVITY,,,LI1;
CNT  ASSIGN, SS(I)=SS(I)+MES(I), 1;
     ACTIVITY,, SS(I).GE.OBJ(I), FIN;
     ACTIVITY/BD5,BT5,SS(I).LT.OBJ(I),BF1;return;
BF1  FREE,TRAC,1;
     ACTIVITY,,, BA1;
RES  RESOURCE,TRACTOR(CAP),BN1;
```

Figure 4- 2 SLAM II network statement of bulldozing atomic model

The SLAM statement differs somewhat from the CYCLONE submodel. Three statement lines starting from "CNT" (counter) in the SLAM II statement do not have counterparts in the CYCLONE model because of CYCLONE's limitations. MES(I) is the measurement of this *r-process* model, and SS(I) is the accumulated production. Statement "CNT ASSIGN, SS(I) = SS(I) + MES(I) , 1;" is used to accumulate the production of the *r-process*. "OBJ" is the objective of the *r-process* model. Statements "ACTIVITY, ,SS(I).GE.OBJ,FIN;" and "ACTIVITY/BD5, BT5, SS(I).LT.OBJ, BF1; return;" are used to control the simulation time of this *r-process* model. If the accumulated production exceeds the objective, this submodel will terminate; otherwise, its operation will continue. The same concepts are used in the textual representation of all other atomic models.

(Author's Note: Since this work focuses on the visible portion of the RBM methodology, the SLAM II textual representations of models are not presented in the main body of this Chapter. They are included in Appendix A.)

4.2.2 Excavators for excavation

When an excavator is excavating, four basic work tasks are involved: load bucket (EX1), swing loaded (EX2), dump (EX3), and swing back (EX4). This atomic model is graphically represented in Figure 4-3. One input port precedes 'load bucket' task. One output port follows 'dump' task.

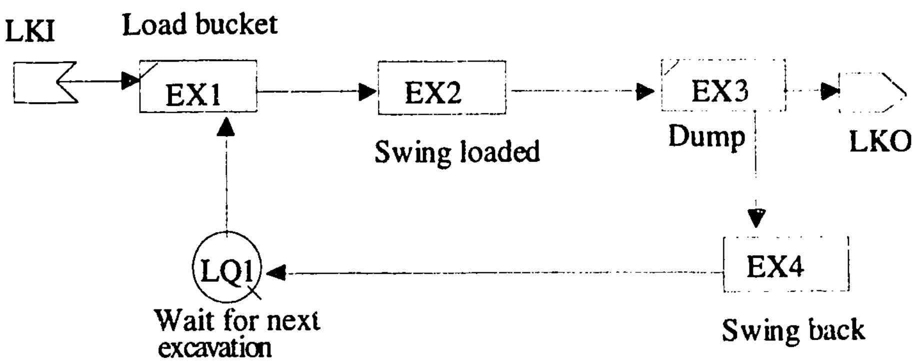


Figure 4- 3 Excavation atomic model

4.2.3 Excavators or loaders for loading

When an excavator or a loader is loading soil to a truck, four basic work tasks are involved: load bucket (LD1), swing loaded (LD2) ("move to dump" for loaders), dump bucket (LD3), and swing empty (LD4) ("move back" for loaders). There are two input ports and one output port in this atomic model as shown in Figure 4-4.

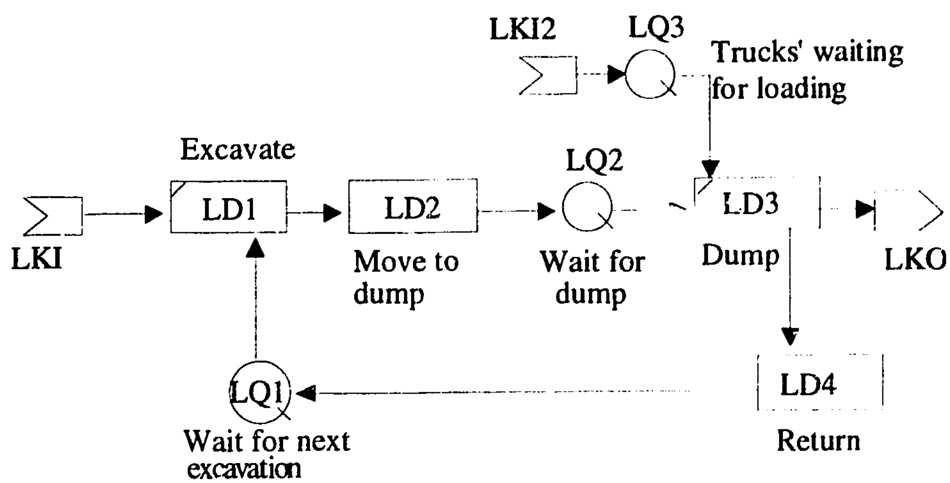


Figure 4- 4 Loading atomic model

4.2.4 Scrapers for scraping

A scraping process has four basic work tasks: load bucket (SC1), haul to dump (SC2), dump (SC3), and return (SC4). Loading often requires the assistance of a pushing tractor. The atomic model is graphically represented in Figure 4-5.

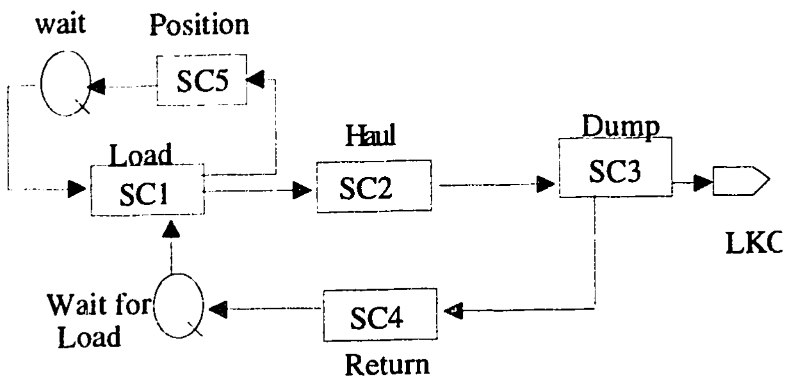


Figure 4- 5 Scraping atomic model

4.2.5 Trucks for hauling

The hauling process of a truck includes four basic tasks: load truck, haul to dump, dump, and return for next cycle. The number of buckets needed to fill a truck is jointly determined by the capacities of the truck and the loading unit. The hauling atomic model is shown in Figure 4-6. The function node 'con' is used to accumulate required number of buckets to fill a truck. However, CYCLONE does not allow variant number of buckets in the model. The SLAM II statement can accomplish this by defining the capacity of hauling fleet and the capacity of the loading units as SLAM II variables, and the number of buckets required to fill a truck will be calculated by using these variables during simulation (truck's capacity divided by the bucket capacity).

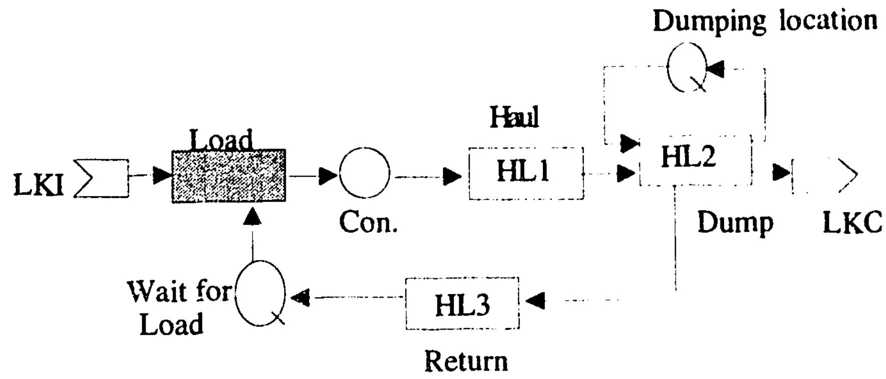


Figure 4- 6 Hauling atomic model

4.2.6 Tractors or graders for spreading

The spreading atomic model, shown in Figure 4-7, is similar to the tractor bulldozing atomic model. When a tractor is used for spreading, the communication ports of the atomic model are different from those in the bulldozing atomic model. The bulldozing atomic model provides output to a loading model. A spreading model may receive input from a hauling model and provide output to a compacting model.

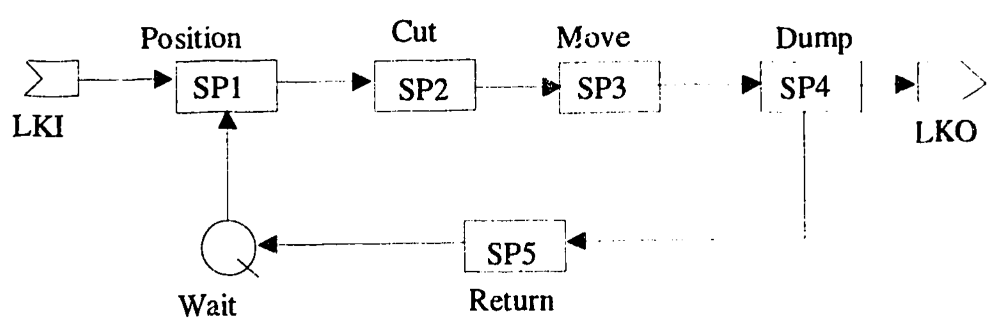


Figure 4- 7 Spreading atomic model

4.2.7 Compactors for compaction

The work tasks involved in a compaction process (Figure 4-8) are: positioning compactor and compaction. Normally the compaction process does not start until the prepared area is large enough to get a proper compaction area.

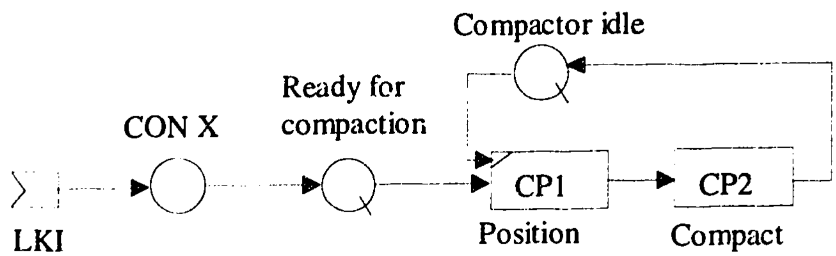


Figure 4- 8 Compaction atomic model

4.2.8 Limitation to the current atomic model library

At this level of development, the environment does not allow the user to add new atomic models to the library. However, a sophisticated user can go to SLAM II to create and add new modular models which have not been covered in the library to the generated entire model. For the purpose of demonstrating the system, however, it is expected that the eight *r-processes* presently provided will meet most requirements of straightforward earthmoving operations - cut, fill, dispose, grade and compact. The continuing development with earthwork contractors will complete the system by including the following:

1. the addition of further atomic models to the library to address a wider range of earthwork operations, and

2. The capacity to allow an average user to create and store new atomic models to the library, with the automated generation of the corresponding SLAM II network statements.

4.3 RESOURCE SPECIFICATIONS

To provide default common resource attributes, an embedded resource library (database) is implemented by including all types of equipment used for general earthmoving projects. Seven types of equipment typical to are included in this resource library: tractors, loaders, excavators, scrapers, trucks, graders, and compactors. The information included in this database is currently limited to the various operating costs for each machine: internal rate, fuel cost, operator cost, support labor cost, and the total cost. The user can create, update, or view the database through a provided user interface. Figure 4-9 shows the screen which facilitates the creation or update of the database. Figure 4-10, as an example, lists the available tractors in the database. The unit for all these operation costs is dollars per hour.

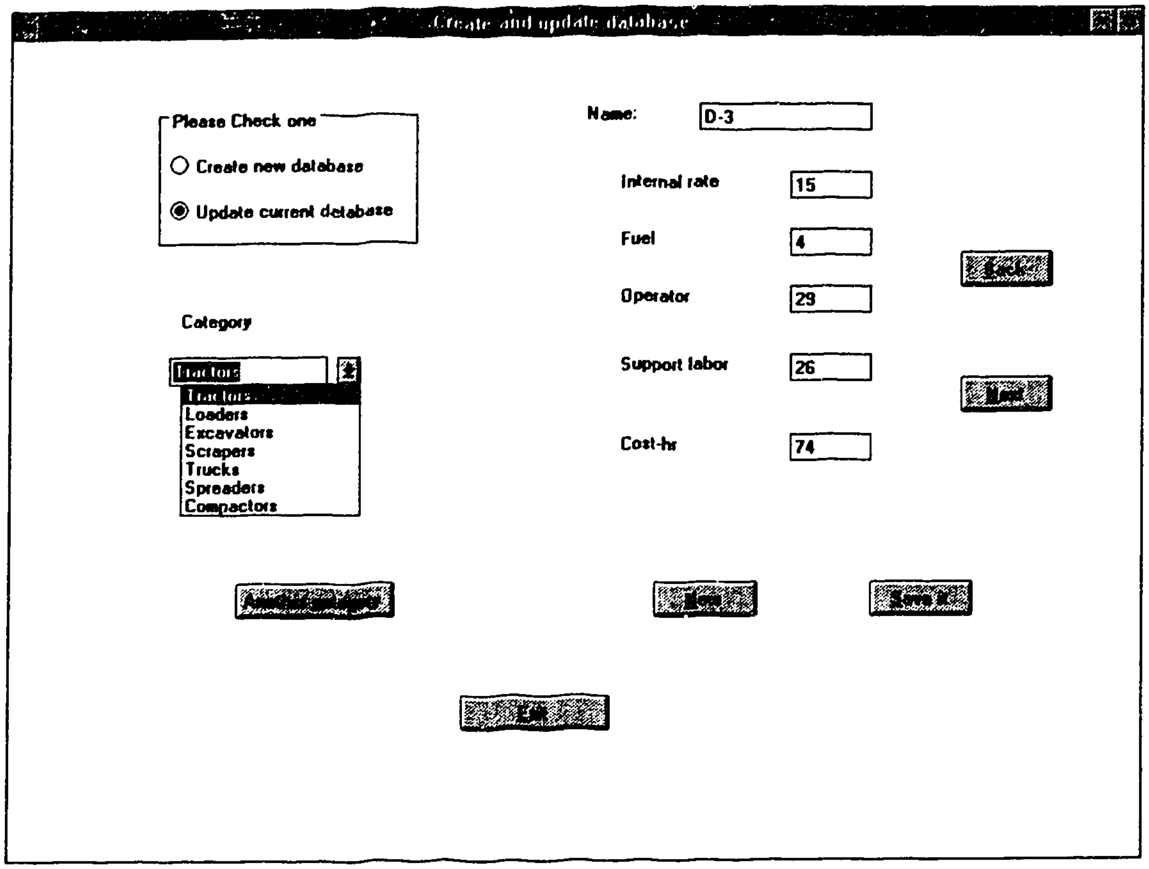


Figure 4- 9 Creating and updating database interface

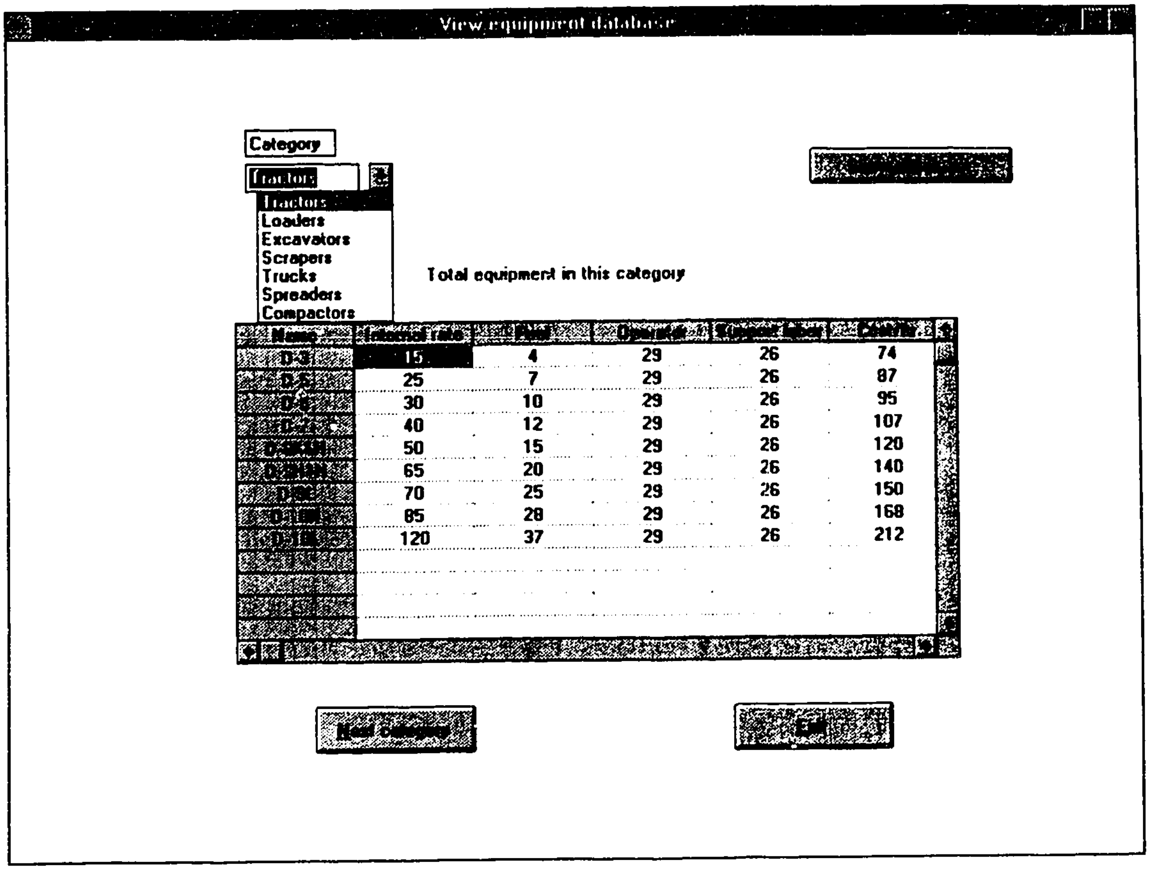


Figure 4- 10 Viewing a database

Resource specification is the process to select the required equipment from the resource library. In addition to the common attributes included in the database, the user must input the project-specific attributes for selected resources. The typical project-specific resource attributes are: operator skill, equipment condition, and required number. Through the interface shown in Figure 4-11, the specific resource is selected, then the attributes are specified. All resources specified by the user together with their attributes are stored to a file named "PROJNAME.RES". Here, PROJNAME is the project name created by the user when initializing a new project.

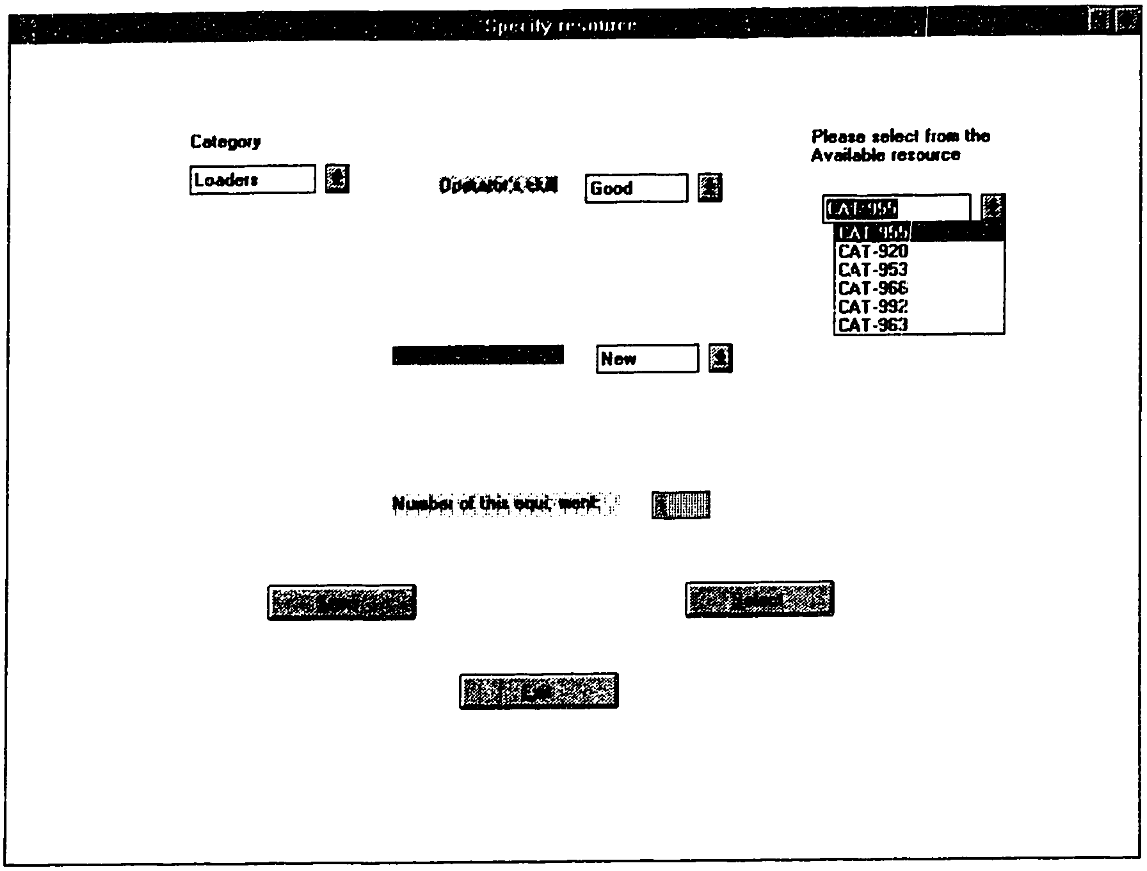


Figure 4- 11 Resource specification interface

4.4 PROJECT SPECIFICATIONS

The project specifications allow the user to describe the formulation of a construction project, assign resources to corresponding *r-processes*, specify project site conditions, specify objectives and units of measurement for defined *r-processes*.

4.4.1 *R-processes* and sequences

An *r-process* has a corresponding atomic model in the model library, and can be executed by one or more resources. Eight basic *r-processes* have been defined in this example earthwork environment. They are bulldozing, excavation, loading, hauling, scraping, assisting, spreading, and compaction. The assisting *r-process* was specifically designed for instances where equipment is used in a supporting mode, such as tractor used for pushing scrapers. A general earthmoving project can be adequately represented by these eight *r-processes.*

Through the project specification interface, the user first inputs the total number of *r-processes* which makeup the construction system. It is then necessary to specify the names of these *r-processes.* Then the user can specify the logical sequences among the specified *r-processes.* All *r-processes* are arranged as a matrix that allows the user to specify the followers of each *r-process.* Checking an *r-process* along its vertical axis indicates that it follows the *r-process(es)* referenced by the horizontal axis. Figure 4-12 shows the *r-process* specification interface and illustrates an example project with six *r-processes.* The logical sequences for this example project are: *r-process* 2 (Loading1) follows *r-process* 1 (Bulldozing1); *r-processes* 3 and 4 (Hauling1 and Hauling2) follow *r-process* 2; *r-process* 5 (Spreading1) follows *r-processes* 3 and 4; *r-process* 6 follows *r-process* 5.

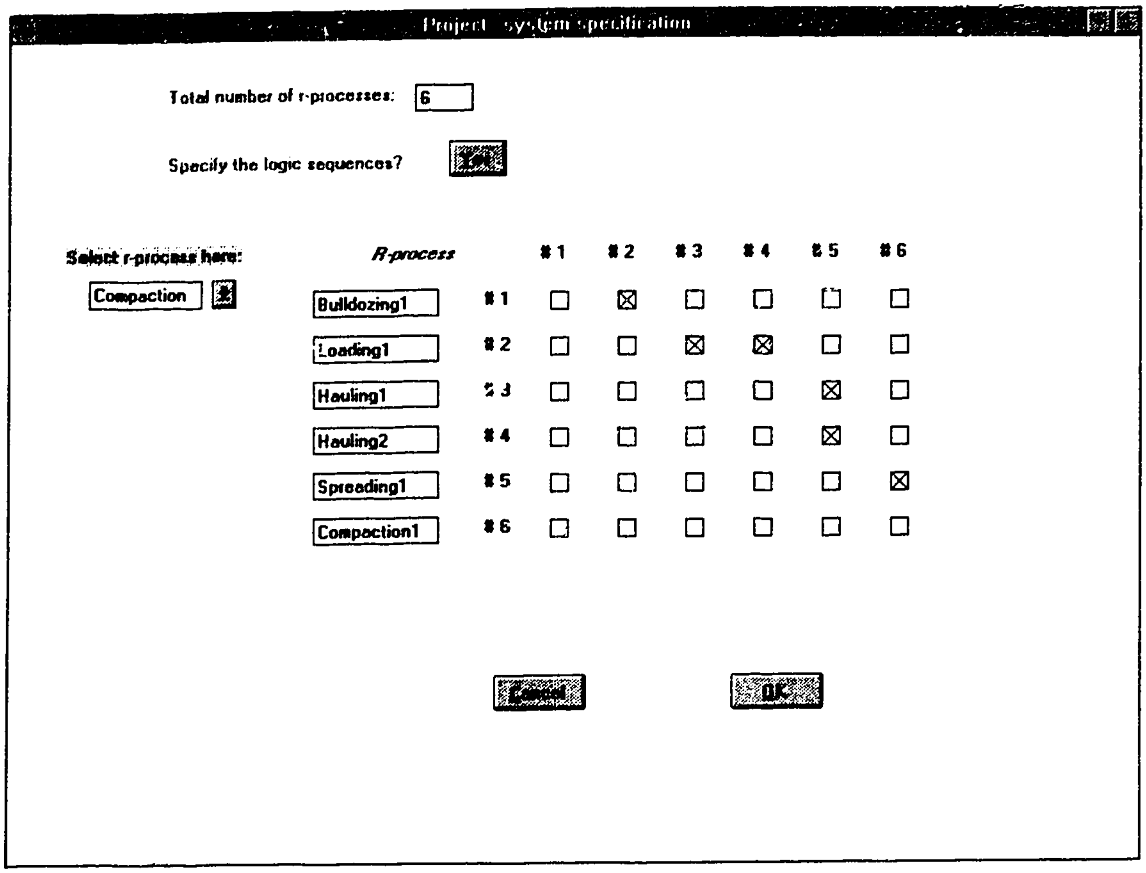


Figure 4- 12 System specification interface

4.4.2 Assignment of resources

Each *r-process* needs at least one resource to execute it. At the same time, each resource can work on at least one *r-process*. In the process of assigning resources, the user is required to associate *r-processes* with resources. For example, TRACTOR D-9L for Bulldozing1, TRACTOR D-6 for Spreading2. An interface shown in Figure 4-13 was designed to facilitate the user to assign resources to *r-processes*.

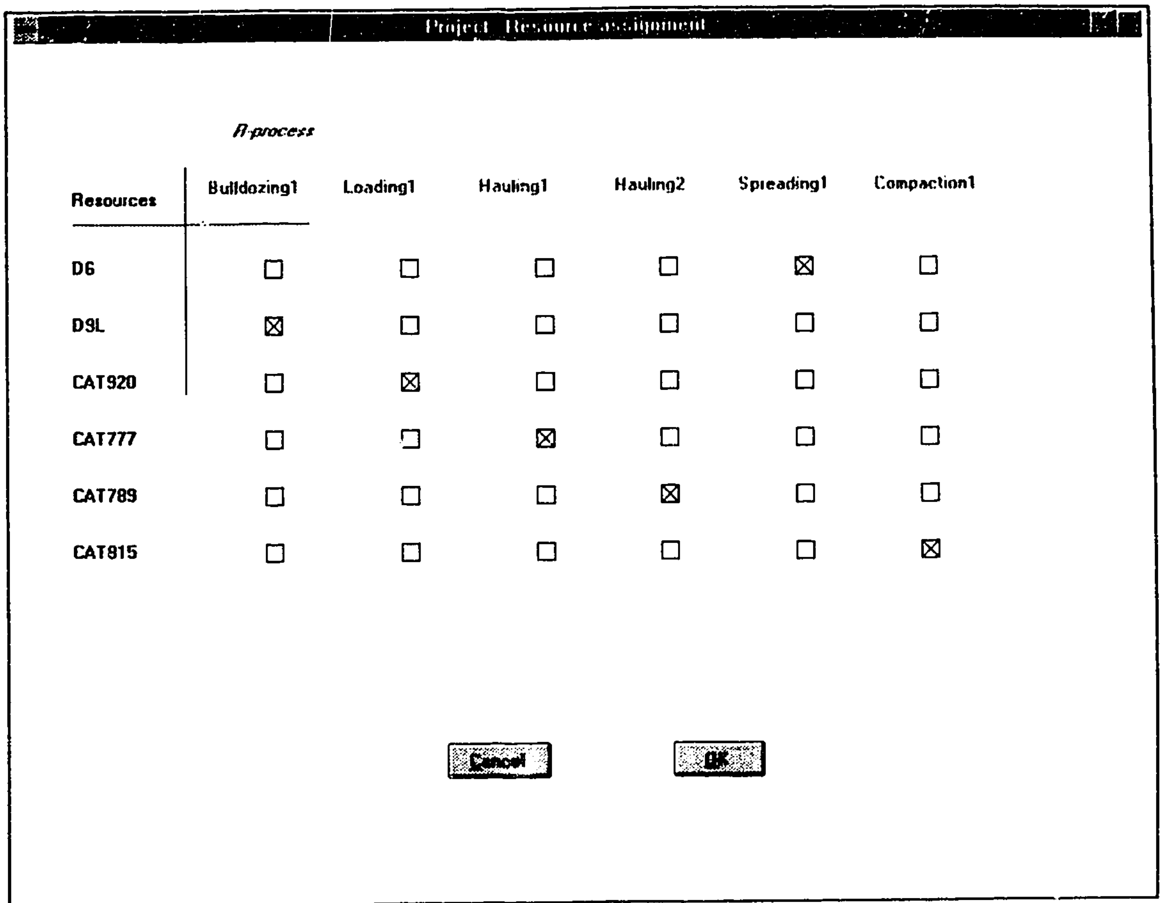


Figure 4- 13 Resource assignment interface

If a single resource is assigned to more than one *r-process*, it will be treated as a shared resource. The user is required to specify priority to these *r-processes*.

4.4.3 Project Site Conditions

Project site conditions affect a construction system through their effect on *r-processes*. For example, road conditions affect hauling *r-processes*. The project site conditions, therefore, can be introduced into the overall system through their influence on individual *r-processes*. At current state, the user is required to convert site conditions into work task durations with stochastic distributions, which can be determined by analyzing historical site data.

Each task duration must be specified with a distribution as well as corresponding parameter values. All distributions that are acceptable to SLAMSYSTEM can be specified. The user can directly enter a task duration in the provided box. The user can also double-click a task duration box. This brings the distribution dialog box (Figure 4-15) into view. The user can select a distribution, enter appropriate parameter values, then click on the OK bottom to transfer the selected duration distribution to a task duration. Distribution functions have to be presented in standard SLAM II syntax if the user selects to directly enter task durations. Otherwise, the syntax is automatically handled by the environment. Table 4-1 lists the SLAM II standard distributions.

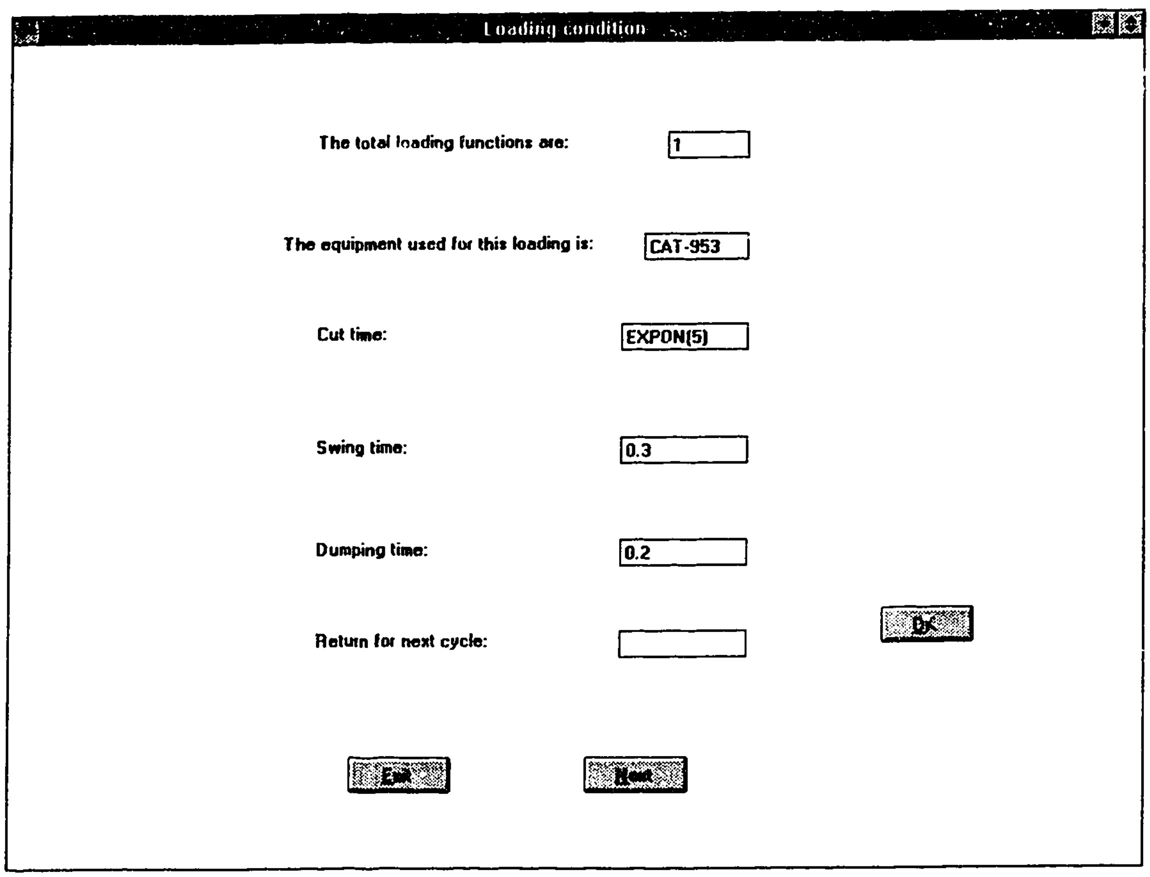


Figure 4- 14 Duration specification interface

Table 4-1 Standard SLAM distributions

Distribution	Variable name	Required parameters
Constant	Constant value	One single value
Exponential	EXPON(mean)	The mean value.
Uniform	UNFRM(ULO,UHI)	ULO -- Low bound of the interval UHI -- High bound of the interval
Weibull	WEIBL(BETA,ALPHA)	BETA -- Scale parameter ALPHA -- Shape parameter
Triangular	TRIAG(XLO,MODE,XHI)	XLO -- Low boundary of the interval XMODE -- mode XHI -- High boundary of the interval
Normal	NORM(XMN,STD)	XMN -- Mean STD -- Standard distribution
Lognormal	RLOGN(XMN,STD)	XMN -- Mean STD -- Standard distribution
Erlang	ERLNG(EMN,XK)	EMN -- Mean XK -- Number of exponential samples
Gamma	GAMA(BETA,ALPHA)	BETA -- Parameter ALPHA -- Parameter
Beta	BETA(THETA,PHI)	THETA -- Parameter PHI -- Parameter
Poisson	NPSSN(XMN)	XMN -- Mean

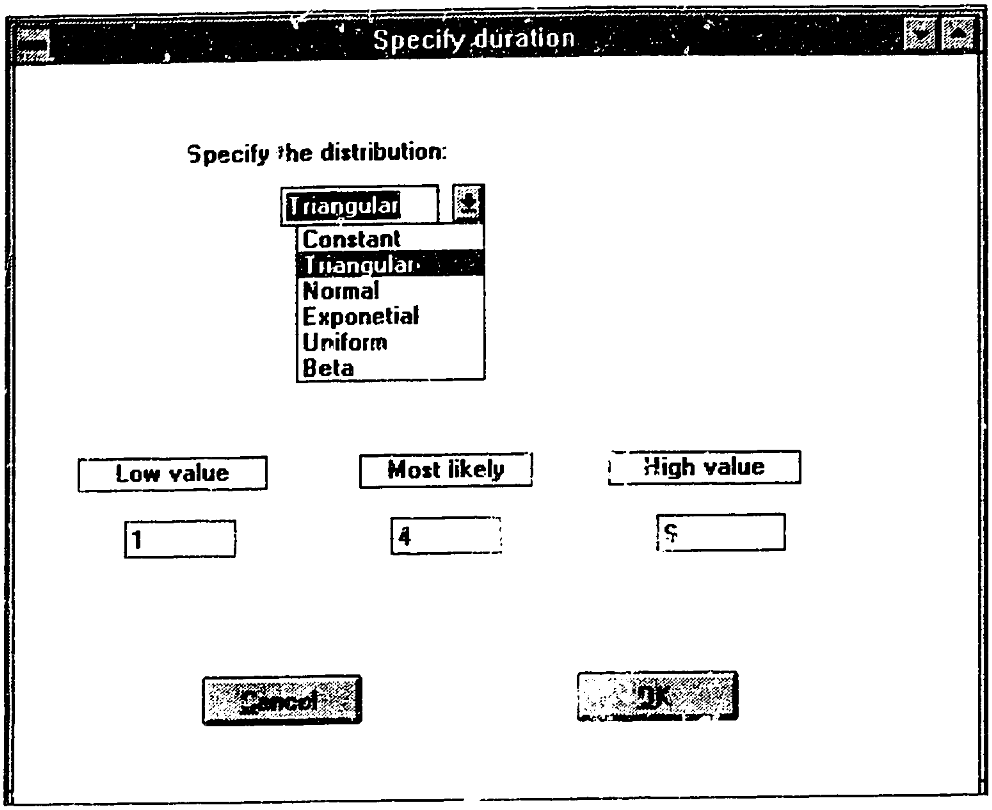


Figure 4- 15 Distribution specification interface

Figure 4-14 illustrates the task durations that must be specified for a loading *r-process*. Task durations that must be specified for the remaining seven *r-processes* are described below.

a. Required task durations for bulldozing *r-process*

Five task durations are required to be specified: 1) positioning the equipment for a new cycle, 2) cut time, 3) move time, 4) dump time, and 5) return time.

b. Required task durations for excavating *r-process*

Four task duration specifications are required: 1) load bucket, 2) swing loaded, 3) dump, and 4) swing back.

c. Required task durations for hauling *r-process*

A hauling *r-process* has three tasks: 1) haul loaded, 2) dump, and 3) return empty. However, the haul loaded and return empty tasks can be further divided into detailed sections to allow for varying grades and road conditions. For example, the traveling process of a truck can be divided into three sections for hauling loaded, and two sections for returning empty to describe specific road and traffic conditions in each section. The user can specify the traveling durations of the truck's round trip in five portions (three for hauling and two for returning). Finally the total hauling/returning time is calculated by adding hauling/returning durations through a user "insert function".

d. Required task durations for scraping *r-process*

Four task durations are required in scraping: 1) load bucket, 2) move to dump, 3) dump, and 4) return for next cycle. If a tractor is used to assist loading, the time required for the tractor to become available for the next push can also be provided.

e. Required task durations for spreading *r-process*

Five duration specifications are required for spreading *r-processes*: 1) positioning the equipment for a new cycle, 2) cut time, 3) move time, 4) dump time, and 5) return time.

f. Required task durations for compacting *r-process*

There are two work tasks in the compaction *r-process*, 1) positioning the compactors, and 2) compacting.

4.4.4 *R-process* measurements and objectives

The user can uses the provided interface (detailed in Appendix C) to input measurements and objectives for all *r-processes*. Measurement is necessary for each *r-*

process. The user can provide an objective for each *r-process* to jointly control the simulation time; however, all *r-processes* do not have to be assigned objectives - the simulation time will be controlled by those which have objectives assigned to them. Moreover, the simulation termination can also be controlled by giving a total simulation time during generating the control file for an application.

4.5 PROJECT-SPECIFIC ATOMIC MODEL GENERATION

After resources, project site conditions and *r-process* objectives have been specified, project-specific atomic generation process will incorporate them into the general atomic models in the atomic model library. Project-specific atomic model generation includes the following tasks: 1) re-label nodes in the *r-process* atomic models, 2) plug in actual data into model statements.

4.5.1 Labeling Nodes

In order to precisely represent the logical sequences of an atomic model in statement, all nodes must be uniquely labeled in SLAM II language. Although all nodes in the atomic models in the library were previously labeled, they must now be re-labeled according to a common convention to avoid duplications.

4.5.1.1 Standard label convention

Because there is only one atomic model for one type of *r-process*, and there might be any number of one type of *r-processes* in a project, the re-labeling convention ensures that each separate occurrence of an atomic model is uniquely identified.

The new label of a node starts with one alpha character and is followed by two numeric characters:

$$T + F + L$$

where T represents the *r-process* type with the following convention:

$$T = \begin{cases} \text{B} & \text{For bulldozing r - process} \\ \text{L} & \text{For loading r - process} \\ \text{H} & \text{For hauling r - process} \\ \text{E} & \text{For excavating r - process} \\ \text{R} & \text{For scraping r - process} \\ \text{S} & \text{For spreading r - process} \\ \text{C} & \text{For compacting r - process} \end{cases}$$

F is a number starting with 0, which allows the *r-process* to be identified separately from other occurrences of the same *r-process* type. For instance, if there are three bulldozing *r-processes*, they are numbered as 0, 1, and 2.

L is the number of a node in a *r-process* model starting with 0. If there are ten nodes in a *r-process* model, for example, they will be numbered as 0, 1, ... 9 respectively.

This labeling convention creates unique labels for every project-specific atomic models in the scope of an entire simulation model as required by SLAM II. For example, the fifth nodes of the first and second loading *r-processes* are labeled as "L05" and "L15" respectively, and the fifth nodes of the first and second hauling *r-processes* are labeled as "H05" and "H15" respectively.

4.5.1.2 Exceptions to the re-labeling convention for communication nodes

There are two exceptions to the re-labeling convention: for input and output communication nodes. In this case, original labels will not be re-labeled during project-specific atomic model generation. Instead, they will be relabeled at the entire model generation stage in the following manners:

a. Input nodes

An input communication node shows information flowing in from another model, and is re-labeled as:

T "I" F N

Where 'T' and 'F' have the same meanings as for the standard re-labeling convention. 'I' donates 'input', and 'N' is the identifying number of this input node in the *r-process* model. For instance, the second input node of the second loading *r-process* is labeled as 'LI12'.

b. Output nodes

An output can be described in SLAM II by an 'ACTIVITY' which can specify the following node. Two types of output nodes can be defined in this RBM environment: direct output and indirect output. A direct output activity is labeled 'DNK' and will directly transfer the simulation entity from an *r-process* model to another. An indirect output is labeled 'LNK' and will pass the simulation entity to a linking structure in which the entities are further processed (as detailed in Chapter 3) and forward to the appropriate input node of a following *r-process*. 'DNK' and 'LNK' labels are unchanged during submodel generation for later reference during entire model generation.

4.5.2 Plug Actual Data into Atomic Models

An atomic model in the library shows the operating process of a resource, and does not contain any actual resource and project-related data. The textual representation of a SLAM II model consists of statement lines. Each line is either an 'ACTIVITY' or a 'NODE' which requires corresponding project-specific information.

4.5.2.1 Description SLAM II nodes

SLAM II was invented by Pritsker, and has been widely accepted by the manufacturing industry. The basic SLAM II nodes have been adopted in this modeling environment, and are described as follows (Pritsker, 1986).

ACCUMULATE, FR, SR, SAVE,M;
Function node to accumulate simulation entities
- FR -- Number for first release
- SR -- Number for subsequent release
- Save -- Save criterion specified as: FIRST, LAST, etc.
- M -- Maximum branches to take.

ACTIVITY/A,DUR,COND,NLBL;
Representing work tasks
- A -- Activity number
- DUR -- Duration
- COND -- Condition
- NLBL -- The label of a node which is the end node of the activity.

ASSIGN,VAR=VALUE,M;
Function node to assign values to SLAM II variables
- VAR -- SLAM II variable

AWAIT(IFL/QC),RES/UR,BLOCK OR BALK (NLBL),M;
A location where simulation entities waiting for available resources to proceed
- IFL -- File number
- QC -- Queue capacity
- RES -- Resource name defined on a resource statement
- UR -- Units required

BATCH,NBATCH/NATRB,THRESH,NATRS,SAVE,RETAIN,M;
Function node to batch multiple entities into one entity
- NBATCH -- Number of batch types
- NATRB -- Attibute for sorting
- THRESH -- Batch size
- NATRS -- Attribute to sum toward THRESH
- SAVE -- Attribute saving criterion
- RETAIN -- Entities to retain

FREE,RES/UF,M;
To free occupied resources at an AWAIT node
- RES -- Resource name previously defined on a resource statement
- UF -- Units to free.

GOON,M;
A continue function node

QUEUE(IFL),IQ,QC,BLOCK or BALK(NLBL),SLBLs;
A location where simulation entitie and resources can waite
IFL -- Integer file number
IQ -- Initial queue length
QC -- Queue capacity
SLBLs -- Labels of following SELECT nodes.

RESOURCE/NUM,RES(CAP),IFLs;
A pool describing a resource
NUM -- Resource number
RES -- Resource name
CAP -- Initial capacity
IFLs -- AWAIT node files

SELECT,QSR,SSR,BLOCK or BALK(NBL),QLBLs;
A decision function node
QSR -- Queue selection rule
SSR -- Server slection rule
QLBLs -- Labels of QUEUE nodes.

TERMINATE,TC;
A sink where simulation entities can be destroied
TC -- Termination count.

UNBATCH,NATRR,M;
A function node to unbatch one sigle entity into multiple entities
NATR -- Attribute defining the batch.

4.5.2.2 Numbering activities

SLAM II requires that all activities in a network be given unique numbers. NUMACT, which is defined as a global integral variable and is initialized as 0, is updated with NUMACT = NUMACT + 1 before its value is assigned to a new activity.

4.5.2.3 Numbering files

In SLAM II, all files including QUEUE and AWAIT must be given unique file numbers. NUMQUE is defined as a global variable, initialized as 0, and updated with NUMQUE = NUMQUE + 1 before its value is assigned to a new file.

4.5.2.4 Modules for updating node statements

There are many types of nodes in SLAM II, and each requires related resource or project-related information to concrete an atomic model into an executable model. A specific module for updating each type of node is implemented. A schematic chart is shown in Figure 4-16 for this updating process. A function calls the "plug-in" module, and the "plug-in" module will then call the corresponding module to update each statement line in the atomic model.

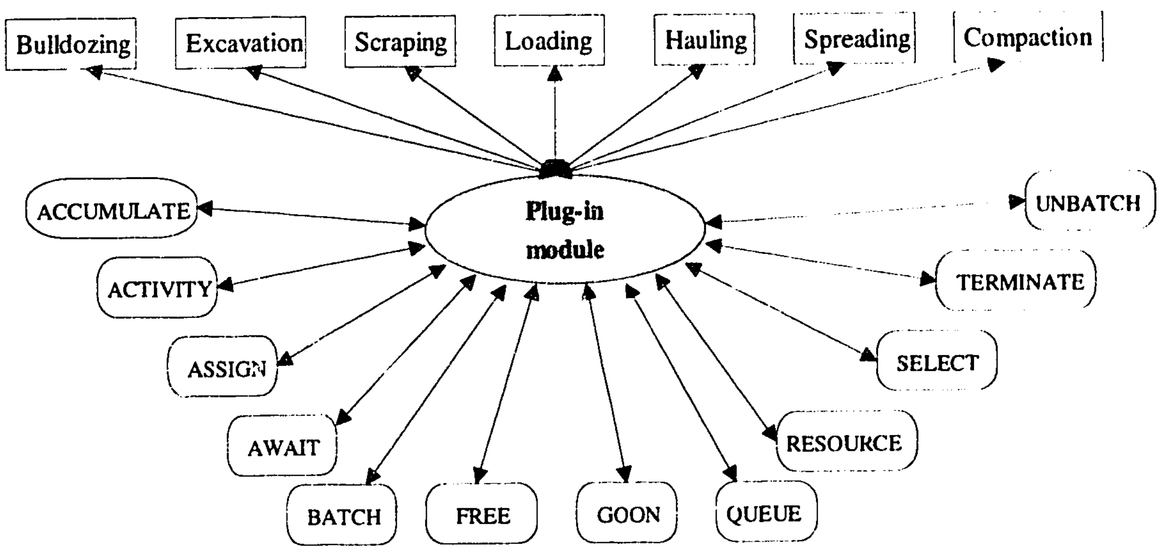


Figure 4- 16 *R-process* model generation

ACCUMULATE module

To update the ACCUMULATE statement, FR (Number of first release) and SR (Number of subsequent release) must be defined according to its *r-process* nature. For example, if Loading1 *r-process* needs six buckets to fill its hauling truck, and Loading2 needs eight buckets to fill its hauling truck, the values of FR and SR in the two cases should be six and eight respectively. A SLAM II variable is defined to automatically handle this during simulation.

ACTIVITY module

Four parameters must be entered in to update an activity statement: activity number, duration, condition, and the label of its ending node. Activity number can be updated as stated in previous section.

The activities in an atomic model are numbered from 1 continuously to the largest number to distinguish each other. For instance, five activities defined in the bulldozing atomic model are specifically associated with actual work tasks: activity 1 as 'positioning machine', activity 2 as 'cut', activity 3 as 'move to dump', activity 4 as 'dump', and activity 5 as 'return for next cycle'. The ACTIVITY module identifies the original activity number and then update the activity duration.

The total hauling time of the hauling fleet is calculated in the user insert (see also section 4.6.3) of SLAM II (Pritsker 1986) according to the number of hauling sections and duration in each section. The same mechanism is used for calculating total return time of hauling units.

The objective of an *r-process* controls its simulation time. The *r-process*'s operation will terminate if its accumulated production has reached its specified objective.

All nodes of an atomic model were originally labeled, and are re-labeled for modeling purposes. Therefore, the ending node label of an activity must be properly updated by associating its old label with its new label.

ASSIGN module

ASSIGN nodes are used to assign values to SLAM II variables. It is necessary to pre-define the meanings of these variables. For example, the capacity of a hauling unit is assigned to a SLAM II ATRIB(I) variable; and the accumulated production of a *r-process* is assign to a SS(I) variable. The ASSIGN module must identify the right SLAM II variables and the right index of array variables by associating with its *r-process*. For example, the capacities of the first and the second hauling fleets are assigned to ATRIB(1) and ATRIB(2) respectively.

AWAIT module

The AWAIT module updates the file number, the resource name as specified in RESOURCE statement, and the required units in the atomic model.

BATCH module

The BATCH statement is used in linking structures. It identifies the right THRESH value according to the *r-process* measurement.

FREE module

The resource name and units to be released should be consistent with the corresponding AWAIT statement.

GOON module

The parameter that should be identified for a GOON statement is the number of following branches which are defined in the original submodel and should remain unchanged.

QUEUE module

The QUEUE module identifies the file number of a queue node, its initial length, queue capacity, and labels of its following SELECT nodes. In the original atomic models, some QUEUE nodes were initialized with 0, and the initial length 0 will be kept in the updated module. If a QUEUE node is used to initialize the number of a resource (e.g. truck number), the initial length will be updated according to the resource specification.

If a QUEUE node is followed by a SELECT node, the label of this SELECT node will be updated accordingly. Otherwise, it will be assigned a default value.

RESOURCE module

The RESOURCE module defines the name and capacity of a resource associated with an AWAIT node and a FREE node. The required locations (file numbers) of AWAIT nodes for a resource should be identified in this RESOURCE module. For shared resources, file numbers will cross multiple *r-process* models.

SELECT module

Selection rules were defined in the original atomic model, and the SELECT module will keep them unchanged. The SELECT module updates the labels of associated QUEUE nodes in the SELECT statement.

TERMINATE module

This module updates the termination counter according to the number of *r-processes* which have been given 'objectives' to control their simulation.

UNBATCH module

This module identifies the correct ATRIB variable to split one entity into multiples. This variable associates with the VAR variable in its previous ASSIGN statement.

4.5.3 Samples of generated project-specific atomic models

Four sample project-specific atomic models including bulldozing, loading, hauling and spreading are generated and shown from Figure 4-17 to Figure 4-20.

```
B00  AWAIT(1), TRACTOR1/1, ;
     ACTIVITY/1, TRIAG(1,2,3), , B01;
B01  GOON,1;
     ACTIVITY/2, UNFRM(2,3), , B02;
B02  GOON,1;
     ACTIVITY/3, RNORM(4,5), , B03;
B03  GOON,1;
     ACTIVITY/4, 1, , B04;
B04  GOON,2;
     ACTIVITY, , , B05;
LNK  ACTIVITY, , , LI1;
B05  ASSIGN, SS(1)=SS(1)+15, 1;
     ACTIVITY, , SS(1).GE.10000, FIN;
     ACTIVITY/5, 3, SS(1).LT.10000, B06;
B06  FREE, TRACTOR1/1, 1;
     ACTIVITY, , , B00;
```

```
RES   RESOURCE, TRACTOR1(1), 1;
```

Figure 4- 17 Generated bulldozing *r-process* model

```
LI01  AWAIT(2), LOADER1/1, ;
      ACTIVITY/6, RNORM(1,3), , L00;
L00   GOON,1;
      ACTIVITY/7, 0.5, , L01;
L01   QUE(3), , 15, BALK(SINK), L03;
LI02  UNBATCH, 1, ;
      ACTIVITY, , , L02;
L02   QUE(4), , 15, BALK(SINK), L03;
L03   SELECT,ASM,POR, , L01,L02;
      ACTIVITY(1)/8, 0.3, , L04;
L04   GOON, 2;
      ACTIVITY, , , L06;
      AC[illegible] L05;
L05   A[illegible],ATRIB(1), ATRIB(1), , ;
DNK   [illegible] HI1;
L06   [illegible]=SS(2)+10, 1;
      A[illegible]SS(2).GE.10000, FIN;
      ACTIVITY/9, 0.4, SS(2).LT.10000, L07;
L07   FREE, LOADER1/1, 1;
      ACTIVITY, , , SINK;
RES   RESOURCE, LOADER1(1), 2;
```

Figure 4- 18 Generated loading *r-process* model

```
HI01  GOON,1;
      ACTIVITY/10, USERF(1), , H00;
H00   AWAIT(5), DUMP1/1, ;
      ACTIVITY/11, 1, , H01;
H01   FREE, DUMP1/1, 2;
LNK   ACTIVITY, , , VI1;
      ACTIVITY, , , H02;
H02   ASSIGN, SS(3)=SS(3)+40, 1;
      ACTIVITY, , SS(3).GE.10000, FIN;
      ACTIVITY/12, USERF(2), SS(3).LT.10000, H03;
H03   QUE(6), , 15, BALK(SINK);
      ACTIVITY, , , H04;
H04   ASSIGN, ATRIB(1)=4, XX(1)=1, ;
DNK   ACTIVITY, , , LI2;
RES   RESOURCE, DUMP1(3), 5;
```

Figure 4- 19 Generated hauling *r-process* model

```
RI01  AWAIT(7), SPREADER1/1, ;
      ACTIVITY/13, TRIAG(1,2,3), , R00;
R00   GOON,1;
      ACTIVITY/14, RNORM(1,2), , R01;
R01   GOON,1;
      ACTIVITY/15, UNFRM(4,7), , R02;
R02   GOON,1;
      ACTIVITY/16, 1, ,RS03;
```

```
R03  GOON,2;
     ACTIVITY, , , R04;
DNK  ACTIVITY, , , CI1;
R04  ASSIGN, SS(4)=SS(4)+8, 1;
     ACTIVITY, , SS(4).GE.10000, FIN;
     ACTIVITY  . SS(4).LT.10000, R05;
R05  FREE, S   R1/1, 1;
     ACTIVIT    INK;
RES  RESOURCE, SPREADER1(1), 7;
```

Figure 4- 20 Generated spreading *r-process* model

4.6 ENTIRE MODEL GENERATION

An entire process (or project) simulation model is obtained by integrating all generated *r-process* models through linking structures. As discussed in Chapter 3, various direct and indirect linking structures can be defined for earthmoving operations. To avoid duplication, only the implemented SLAM II statement for each linking structure is illustrated in this section. The reader can refer Section 3.9 in Chapter 3 for the detailed descriptions and graphic representations for those linking structures.

4.6.1 Direct linking structures

a. One-one link

```
ACTIVITY, , , I1;
```

Figure 4- 21 One-one direct linking

Where I1 is the label of the input port of submodel 1. The start of this linking activity is the output port of submodel A.

b. One-multiple link with all branch releases

```
     ACTIVITY, , , GON;
GON  GOON, M;
     ACTIVITY, , ,I1;
```

```
        ACTIVITY, , ,I2;
        .
        .
        .
        ACTIVITY, , , IM;
```
Figure 4- 22 One-multiple direct link

Where M is the number of the following submodels. I1, I2, ..., IM are labels of the input ports of Submodel 1, Submodel 2, ..., Submodel M respectively.

<u>***c. One-multiple link with one branch release***</u>

```
        ACTTIVITY, , , SLC;
SLC     SELECT,QSR,Q1,Q2, ...QM;
Q1      QUEUE,0, , ,SLC;
        ACTIVITY, , , I1;
Q2      QUEUE,0, , ,SLC;
.       ACTIVITY, , , I2;
.       .
.       .
        .
QM      QUEUE,0, , ,SLC;
        ACTIVITY, , , IM;
```
Figure 4- 23 One-multiple direct link

Where M is the number of the following submodels. I1, I2, ..., IM are labels of the input ports of Submodel 1, Submodel 2, ..., Submodel M respectively. QSR is the queue selection rule. In our implementation, we use CYClic priority rule.

<u>***d. Multiple-one with multiple releases***</u>

```
        ACTIVITY, , , IA;
        ACTIVITY, , , IA;
        .
        .
        .
        ACTIVITY, , , IA ;
```
Figure 4- 24 Multiple-one link with multiple releases

e. Multiple-one link with one release

```
        ACTIVITY,,,Q1;
Q1      QUE,,,ACC;
        ACTIVITY,,, Q2;
Q2      QUE,,,ACC;
        .
        .
        ACTIVITY,,,QM;
QM      QUE,,,ACC;
ACC     SELECT,ASM,,Q1,Q2,...,QM;
        ACITVITY,,, IA;
```

Figure 4- 25 Multiple-one with multiple releases link

After the SELECT node collects one entity from its preceding submodel and then releases one entity to its following submodel.

4.6.2 In-direct linking structures

a. One-one indirect link

```
        ACTIVITY,,, UB;
UB      UNBATCH,X,1;
        ACTIVITY,,, QE;
QE      QUEUE,0,,;
        ACTIVITY,,, BA;
BA      BATCH,,,Y,,;
        ACTIVITY,,, IB;
```

Figure 4- 26 One-one indirect link

Where IB is the label of the input port of submodel 1. The start of this linking activity is the output port of submodel A. The values of X and Y are calculated according to the specified measurements for submodel A and submodel B respectively.

b. One-multiple indirect link

```
        ACTIVITY,,,UB;
UB      UNBATCH,X,I;
        ACTIVITY,,,QE;
QE      QUEUE,0,,;
        ACTIVITY,,,BA1;
        ACTIVITY,,,BA2;
        .
        .
        .
        ACTIVITY,,,BAM;
BA1     BATCH,,,Y1,,;
        ACTIVITY,,,I1;
BA2     BATCH,,,Y2,,;
        ACTIVITY,,,I2;
        .
        .
        .
BAM     BATCH,,,YM,,;
        ACTIVITY,,,IM;
```

Figure 4- 27 One-multiple indirect link

Where M is the number of the following submodels. I1, I2, ..., IM are labels of the input ports of Submodel 1, Submodel 2, ..., Submodel M respectively. Y1, Y2, ..., YM are the measurements of submodel1, 2, ..., M respectively.

c. Multiple-one indirect link

```
        ACTIVITY,,,UB1;
UB      UNBATCH,X1,1;
        ACTIVITY,,,QE;
        ACTIVITY,,,UB2
UB      UNBATCH,X2,1;
        ACTIVITY,,,QE;
        .
        .
        .
        ACTIVITY,,,UBM;
UB      UNBATCH,XM,1;
        ACTIVITY,,,QE;
QE      QUEUE,0,,;
        ACTIVITY,,,BA;
BA      BATCH,,,Y,,;
        ACTIVITY,,,IB;
```

Figure 4- 28 Multiple-one indirect link with multiple releases

4.6.3 Embedded knowledge for identifying required links

The various linking structures are designed for coupling purposes. Which one fits a specific situation is determined by the involved *r-process*. The RBM environment surveys the input and output communication ports for all *r-processes*. The preceding (or following) *r-processes* of an input (or output) port can be obtained according to the user specified logic sequence among the *r-process* of a project. Whether a "one-one", "multiple-one", or "one-multiple" link should be used can then be determined by the number of involved *r-processes*. For instance, if an *r-process* is followed by another three *r-processes*, then a one-multiple (three) link should be used.

Should a direct link or an indirect link be used depends on the types of involved *r-processes*. The link required between bulldozing and loading is different from that required between loading and hauling. Based on the characteristics of earthmoving operations, the knowledge embedded in the RBM environment is designed as follows.

Both bulldozing and scraping *r-processes* can be linked to loading or spreading *r-processes* using an indirect link. Between loading and hauling, a truck entity is first divided into multiple entities based on the ratio between the capacities of the truck and the loading unit (truck's capacity/loader's capacity); a direct link then route those entities to the loading *r-process*; the same amount of entities are combined into one entity which directly flows to the hauling *r-process* through another direct link.

An excavation *r-process* can also be linked with hauling *r-process* following the same rule as between loading and hauling if the excavation is for loading purposes. An excavation *r-process* can also be linked with spreading *r-process* through indirect links. The links between hauling and spreading or between spreading and compaction are indirect.

4.6.4 Sample of an Entire Model

For a project with four *r-processes* including bulldozing, loading, hauling and spreading, the entire generated SLAM II model is shown in Figure 4-29, and its corresponding SLAM II network is shown in Figure 4-30.

```
         RESOURCE, SPREAD1(1), 7;
         RESOURCE, DUMP1(3), 5;
         RESOURCE, LOADER1(1), 2;
         RESOURCE, TRACTOR1(1), 1;
; BULLDOZING R-PROCESS
B00   AWAIT(1), TRACTOR1/1, ;
      ACTIVITY/1, TRIAG(1,2,3), , B01;
B01   GOON,1;
      ACTIVITY/2, UNFRM(2,3), , B02;
B02   GOON,1;
      ACTIVITY/3, RNORM(4,5), , B03;
B03   GOON,1;
```

```
        ACTIVITY/4, 1, , B04;
B04     GOON,2;
        ACTIVITY, , , B05;
        ACTIVITY, , , BD1;
B05     ASSIGN, SS(1)=SS(1)+15, 1;
        ACTIVITY, , SS(1).GE.10000, FIN;
        ACTIVITY/5, 3, SS(1).LT.10000, B06;
B06     FREE, TRACTOR1/1, 1;
        ACTIVITY, , , B00;
; LINKING PARTS
BD1     ASSIGN, ATRIB(21)=15;
        ACTIVITY, , , ;
        UNBATCH, 21, 1;
        ACTIVITY, , , ;
BD11    SELECT, CYC, POR, , QBD1;
QBD1    QUE(8), , , ;
        ACTIVITY, , , ;
        BATCH, ,10, , LAST, , 1;
        ACTIVITY, , , LI01;
; LOADING R-PROCESS
LI01    AWAIT(2), LOADER1/1, ;
        ACTIVITY/6, RNORM(1,3), , L00;
L00     GOON,1;
        ACTIVITY/7, 0.5, , L01;
L01     QUE(3), , 15, BALK(SINK), L03;
LI02    UNBATCH, 1, ;
        ACTIVITY, , , L02;
L02     QUE(4), , 15, BALK(SINK), L03;
L03     SELECT,ASM,POR, , L01,L02;
        ACTIVITY(1)/8, 0.3, , L04;
L04     GOON, 2;
        ACTIVITY, , , L06;
        ACTIVITY, , , L05;
L05     ACCUMULATE,ATRIB(1), ATRIB(1), , ;
        ACTIVITY, , , HI01;
L06     ASSIGN. SS(2)=SS(2)+10, 1;
        ACTIVITY, , SS(2).GE.10000, FIN;
        ACTIVITY/9, 0.4, SS(2).LT.10000, L07;
L07     FREE, LOADER1/1, 1;
        ACTIVITY, , , SINK;
; HAULING R-PROCESS
HI01    GOON,1;
        ACTIVITY/10, USERF(1), , H00;
H00     AWAIT(5), DUMP1/1, ;
        ACTIVITY/11, 1, , H01;
H01     FREE, DUMP1/1, 2;
        ACTIVITY, , , HL1;
        ACTIVITY, , , H02;
H02     ASSIGN, SS(3)=SS(3)+40, 1;
        ACTIVITY, , SS(3).GE.10000, FIN;
        ACTIVITY/12, USERF(2), SS(3).LT.10000, H03;
H03     QUE(6), , 15, BALK(SINK);
        ACTIVITY, , , H04;
H04     ASSIGN, ATRIB(1)=4, XX(1)=1, ;
```

```
      ACTIVITY, , , LI02;
; LINKING PARTS
HL1   ASSIGN, ATRIB(22)=40;
      ACTIVITY, , , ;
      UNBATCH, 22, 1;
      ACTIVITY, , , ;
HL11  SELECT, CYC, POR, , QHL1;
QHL1  QUE(9), , , ;
      ACTIVITY, , , ;
      BATCH, ,8, , LAST, , 1;
      ACTIVITY, , , VI01;
; Spreading R-PROCESS
VI01  AWAIT(7), SPREADER1/1, ;
      ACTIVITY/13, TRIAG(1,2,3), , V00;
V00   GOON,1;
      ACTIVITY/14, RNORM(1,2), , V01;
V01   GOON,1;
      ACTIVITY/15, UNFRM(4,7), , V02;
V02   GOON,1;
      ACTIVITY/16, 1, , V03;
V03   GOON,2;
      ACTIVITY, , , V04;
      ACTIVITY, , , SINK;
V04   ASSIGN, SS(4)=SS(4)+8, 1;
      ACTIVITY, , SS(4).GE.10000, FIN;
      ACTIVITY/17, 3, SS(4).LT.10000, V05;
V05   FREE, SPREADER1/1, 1;
      ACTIVITY, , , SINK;
SINK  TERMINATE, ;
FIN   TERMINATE, 4;
      END;
```

Figure 4- 29 Generated SLAM II network statement

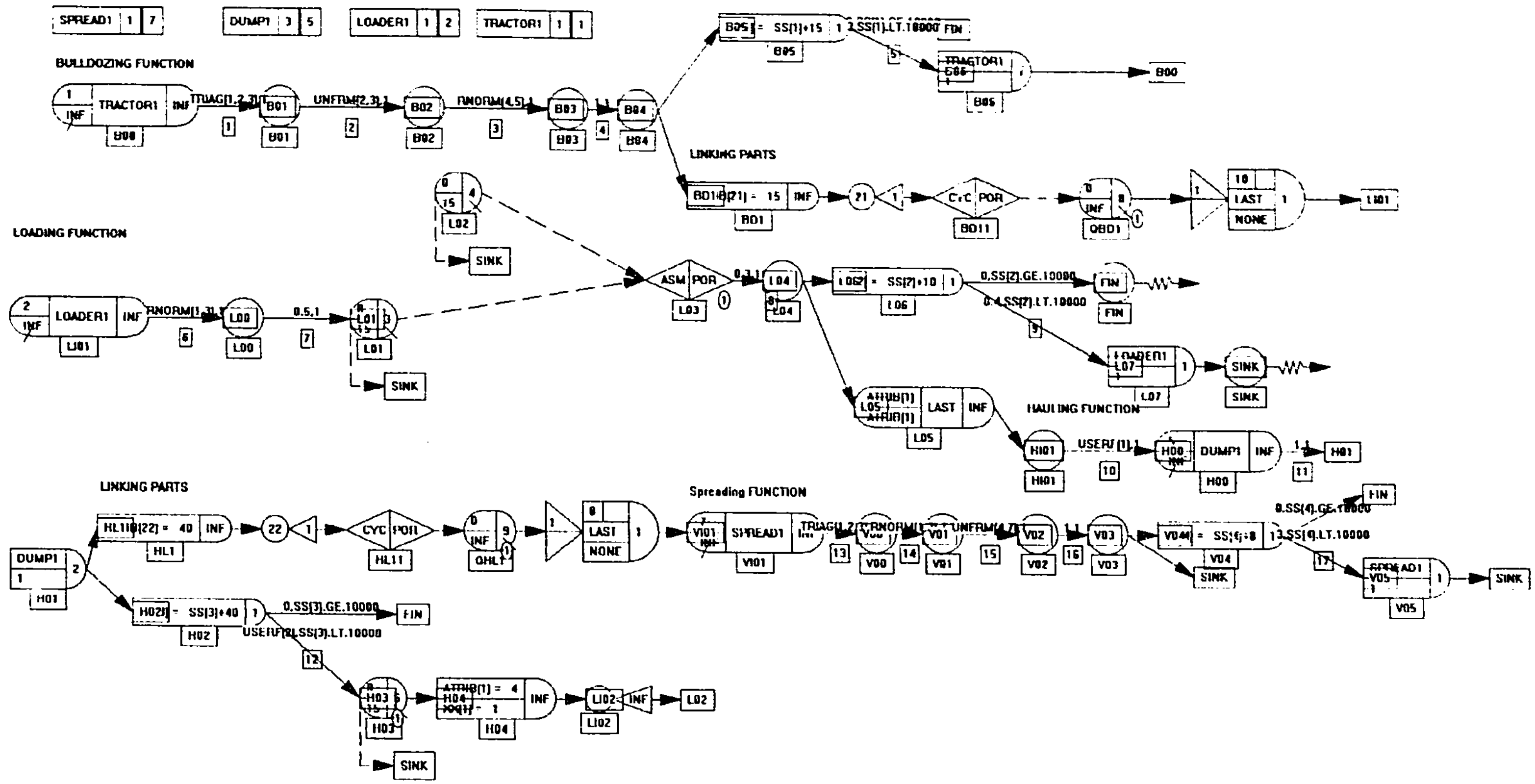


Figure 4- 30 Generated SLAM II graphical network

4.6.5 Control Statement

SLAM II requires a control statement to define a simulation model, including the maximum number of files, number of attributes, simulation time, and others. This RBM environment will generate a control file for an application. In this control statement, simulation entities are automatically initialized and the statistics for the productions of all *r-processes* are collected. For the example model of Figures 4-29 and 4-30, the automatically generated control statement is shown in Figure 4-31.

```
GEN,Jingsheng Shi,proj5,10/27/94,1,Y,Y,Y/Y,Y,Y/1,72;
LIMITS,9,30,100;
ENTRY/1,10,1,1,1,1;
TIMST,SS(1),SS1;
TIMST,SS(2),SS2;
TIMST,SS(3),SS3;7
TIMST,SS(4),SS4;
NETWORK;
INITIALIZE,,,Y;
FIN;
```

Figure 4- 31 Generated control statement

4.6.6 User Insert

The 'user insert' is an extension to SLAM II standard function. In a user insert, the user can write a specific FORTRAN code to solve a specific problem which is not within the standard capabilities of SLAM II. As described in Section 4.4.3.d, the hauling/returning process of earthmoving may be divided into multiple sections. The duration of an activity in SLAM II does not allow the expression of an addition of multiple distributions, instead, it can be represented by a user insert function expression "USERF(I)". Whenever a simulation entity encounter this duration function during simulation, a corresponding user insert function "FUNCTION USERF(I)" will be called. The total hauling/returning time is calculated in this user insert function for multiple hauling/returning sections. This user insert function will be automatically generated if the user specifies multiple hauling/returning sections.

```
      FUNCTION USERF(I)
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      GOTO(1,2), I
    1 USERF=TRIAG(4,6,9)+TRIAG(3,6,8)
      RETURN
    2 USERF=TRIAG(2,4,6)+TRIAG(2,4,6)
      RETURN
      END
```

Figure 4- 32 Generated SLAM II user insert file

4.6.7 Scenario file

A SLAM II scenario file ties all parts of a simulation project together. This file includes the control statement, network statement, and user insert. A scenario file will also be automatically generated for a project and is illustrated in figure 4-33 for the sample project.

```
Scenario:
proj5

Control:
proj5

Network:
proj5

Script:

Facility:

User Insert:
proj5

Notes:

Data:

Curchange:
0
```

Definition:

Figure 4- 33 SLAM II scenario file

4.7 A SAMPLE APPLICATION

To illustrate the RBM concept and the implemented environment, consider a surface mining reclamation project (Shi and Wales 1994) which requires moving 10,000 m^3 soil from one location to another. The major operations involved are: a CAT-D11N dozer pushes soil into a pile, a Hitachi EX-1800 backhoe excavator loads soil from the prepared pile into a CAT-777 truck, trucks move the soil from the cut location to the dump area, a CAT-D9N dozer spreads the soil at dump area, a CAT-16D grader maintains the hauling road. The resource crew and its cost data are listed in Table 4-2. The project manager is interested in finding out the number of trucks that result in the minimal unit cost for this project.

Table 4-2 Cost data of the construction crew

Item (1)	Rate per Hour (2)
Supervision	
1 Foreman (Cut Location)	$39.60
1 Foreman (Dump Location)	$39.60
1/2 time General Foreman (Site)	$19.80
Production Equipment	
EX1800 backhoe	$170.60
CAT-777 trucks	$115.60
D11N (Cut Location) dozer	$155.60
D9N (Dump Location) dozer	$103.60
Support Equipment	
16G Grader (Haul road)	$70.60

The user can specify the resource crews through the interface as shown in Figure 4-34. The user clicks the sequence number (the first column in Figure 4-34) to activate the resource library shown in Figure 4-35, from which the required resources can be selected. The number of resources is specified by the user. The values of the internal rate, fuel cost, operator cost, support labor cost for each piece of equipment are obtained from the embedded database. The user can modify these values on the screen shown in Figure 4-34 to override the specified values for a specific project. The subtotal and total values will be automatically updated when a change is made.

Build-up Crew

Please build-up your crew in this table

Sequence	Name	Number	Internal rate	Fuel	Operator	Support labor	Subtotal
1	Tractors-D9N	1	67	15	21.6	0	103.6
2	Tractors-D11N	1	110	24	21.6	0	155.6
3	Excavators-EX1800	1	117	32	21.6	0	170.6
4	Trucks-CAT777	5	69	25	21.6	0	578
5	Grader-16G	1	35	14	21.6	.	70.6
6							
16	General foreman	0.5	39.6				19.8
17	Foreman	2	39.6				79.2
18	Superintendent	0	52				0
Total			773	210	194.4	0	1177.4

OK

Figure 4- 34 Specify crew through user interface

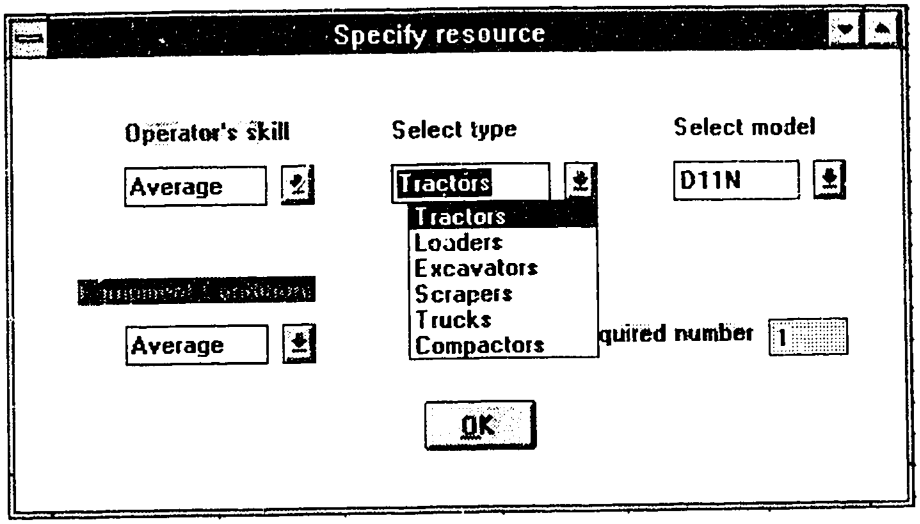


Figure 4- 35 Select resource from resource library

The CAT-16G grader and foremen are assisting resources because they do not directly affect production. To maximize production, the management always allocates the D11N dozer to prepare a soil pile where the EX-1800 is located. Therefore, D11N can also be treated as an assisting resources for it does not directly impact production. Three production *r-processes* including loading, hauling, and spreading can be defined to model this construction project. The sequence of the three *r-processes* is as follows: loading is followed by hauling which is followed by spreading. The resources can be assigned to *r-processes* using the screen shown in Figure 4-36, which shows EX1800 for loading, CAT-777 trucks for hauling, and CAT-D9N for spreading.

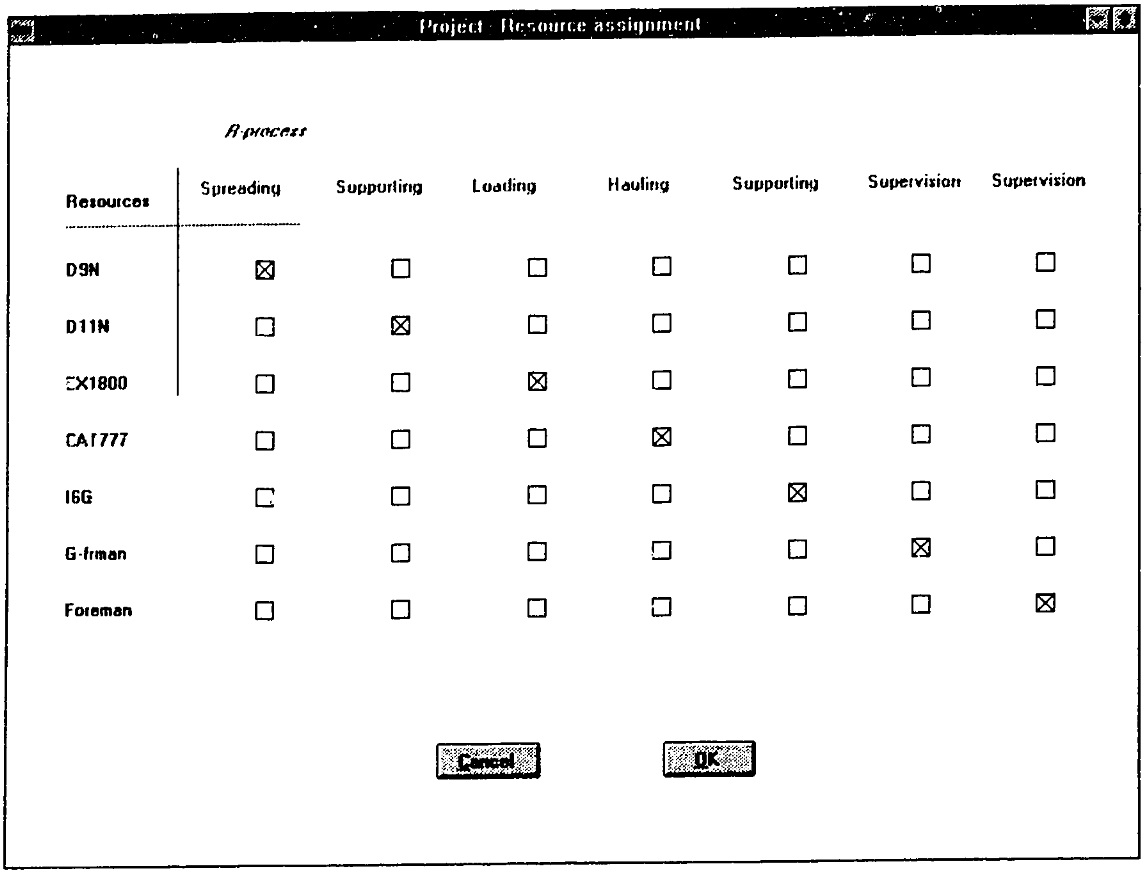


Figure 4- 36 Resource assignment screen

Local environments are specified with the user providing relevant time parameters required for specified resources to conduct given *r-process*. Figure 4-37 is a typical screen for specifying the hauling times with two loaded hauling sections, two empty return sections, and one dump location situation for this sample project. The specifying screens are similar for other *r-processes*. *The system objective* is to spread 10,000 m^3 of excavated soil at the dump area.

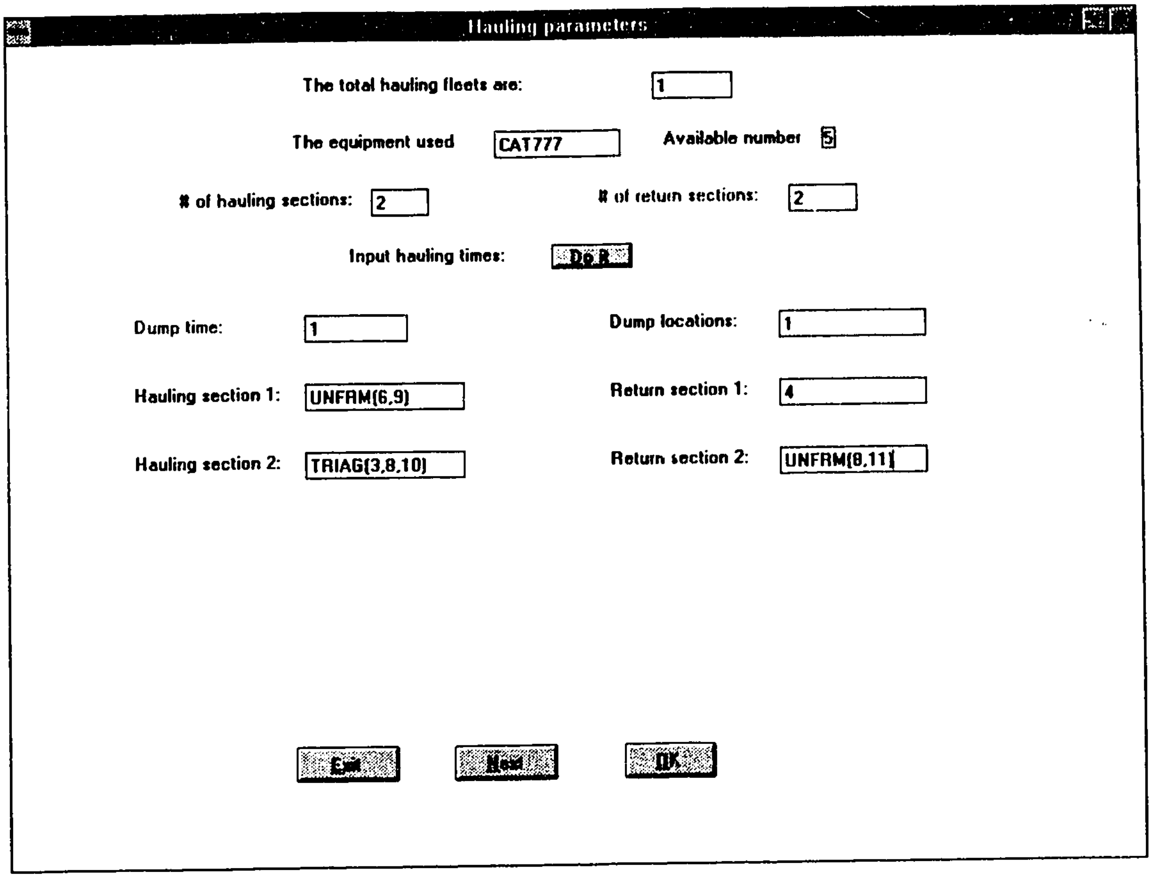


Figure 4- 37 Screen for specifying hauling time parameters

After the resource and project information is specified by the user through provided interfaces, modeling functions become available. The user would first select "submodel generation function" which incorporates specified resource and project information into corresponding atomic models to generate project-specific atomic models. As an example, the hauling *r-process* model generation screen is shown in Figure 4-38 in which a CYCLONE graphical atomic model and the task durations obtained from local environment specification are shown for final examination. The CYCLONE model gives the user a general idea of the atomic model. After the user clicks "OK", a hauling *r-process* model will be generated.

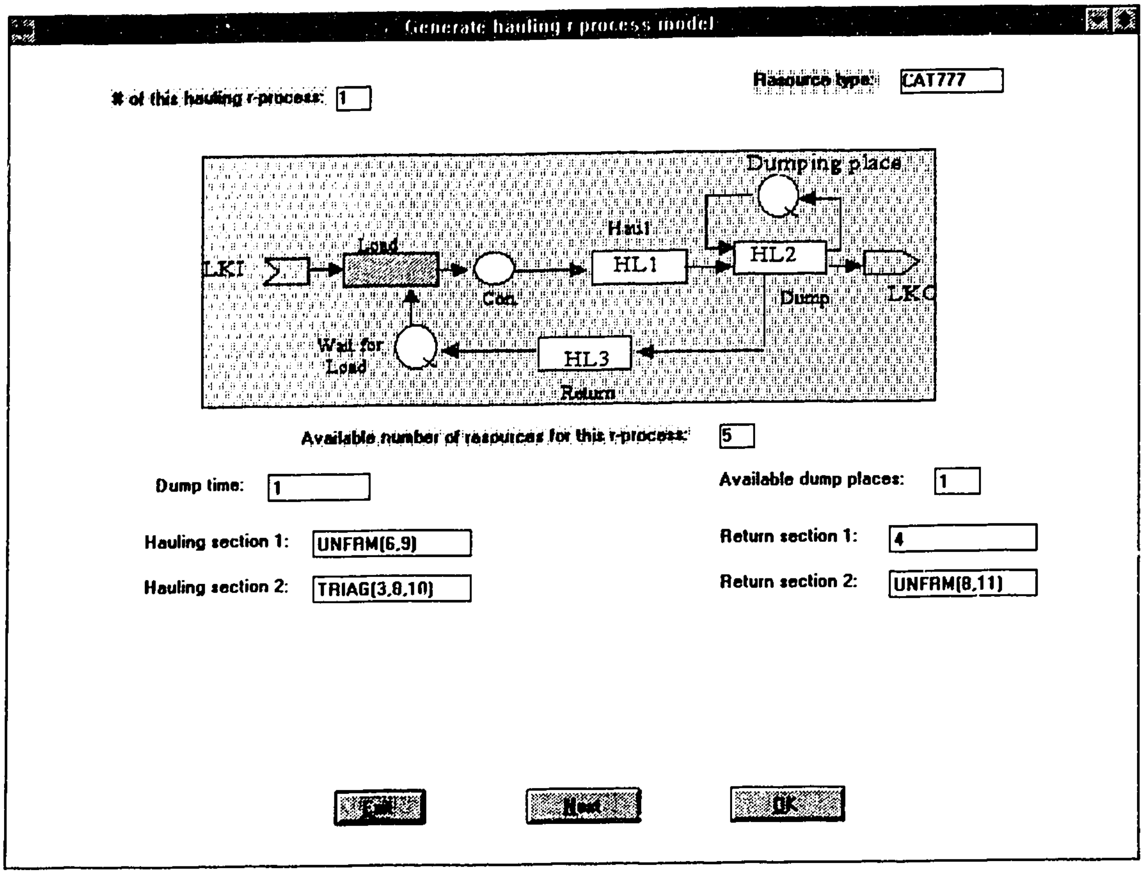


Figure 4- 38 *R-process* model generation screen

After all *r-process* models have been generated, the system automatically assembles these project-specific models into an entire working model by using the appropriate linking structures. The production for each resource in an earthmoving operation is measured by volume (e.g. m^3). In this example, the EX-1800 can load 5 m^3 in each bucket, a CAT-777 truck can haul 40 m^3 soil in each trip, and the D9N dozer can spread 10 m^3 soil in each cycle. Those are the measurements defined for the three *r-processes*. In the embedded base for identifying linking structures, one entity in the hauling *r-process* is divided into 8 entities (= 40/5). Then a one-one direct link is used to link loading and hauling *r-processes*. Before spreading, one entity in the hauling *r-process* is divided into 40 entities, and every 20 of them will be combined into one before flowing to the spreading *r-process*. A one-one indirect link between hauling and spreading is used. The process identifying linking structures is embedded in the RBM environment according to the earthmoving characteristics and does not need any involvement from the user.

The automatically generated SLAM II model for this example is shown in Figure 4-40. Experimenting with this model using SLAMSYSTEM, the production and the unit cost for this construction system under a certain number of trucks can be obtained. Multiple simulation scenarios obtained by changing the number of truck, produce the results shown in Figure 4-39. The graph shows that four trucks result in the lowest unit cost for this project.

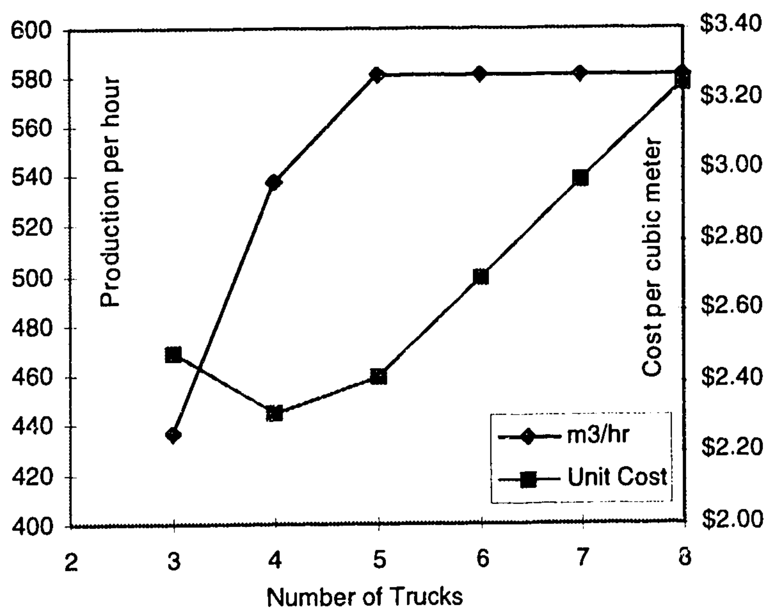


Figure 4- 39 Simulation results for the reclamation project

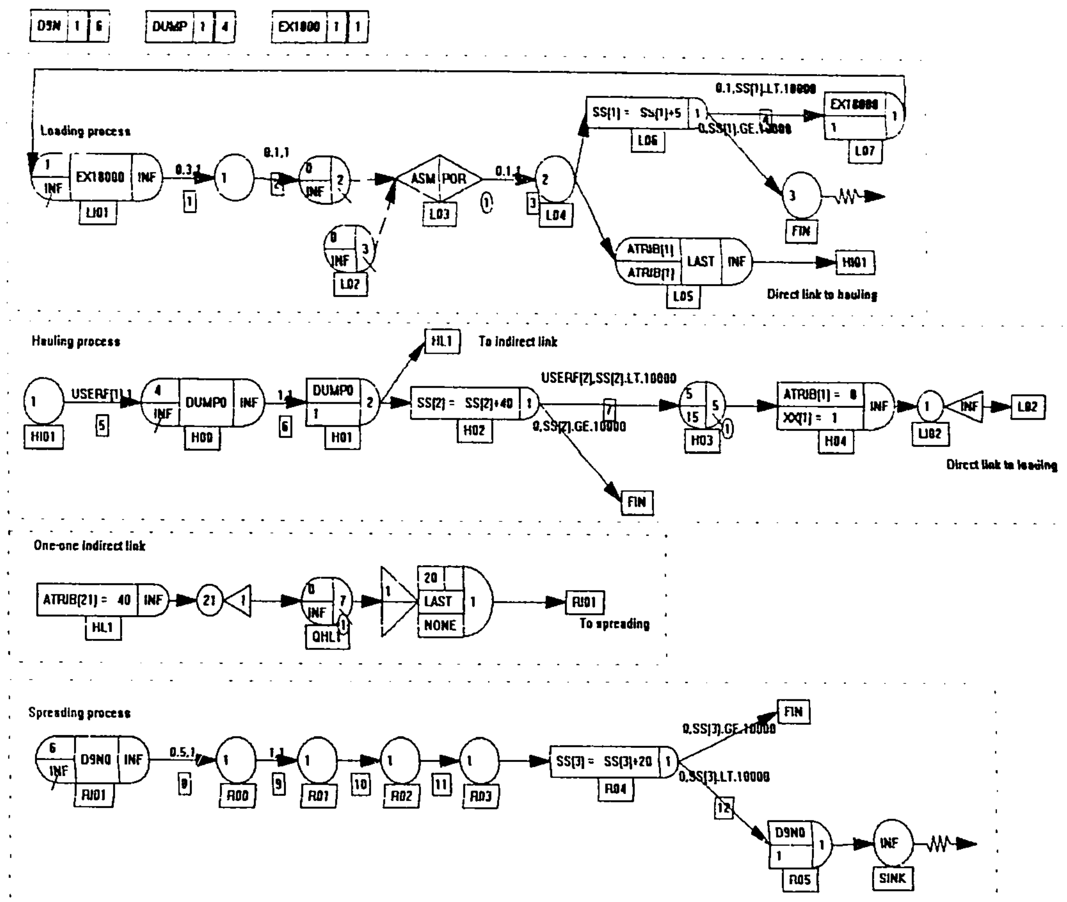


Figure 4- 40 Generated SLAM II model for the earthmoving example

4.8 EMBELLISHMENT FOR THE SAMPLE APPLICATION

If the project manager is interested in adding the "soil preparation" process and another "loading" process to the above simulation model, he/she has to go over the resource specifications again. This can be accomplished by adding one more resource (e.g. a HITACHI EX1100). In the project specification, the user should add two more *r-processes* (a bulldozing *r-process* and another loading *r-process*). The D11N is assigned to "bulldozing", and the EX1100 is assigned to "loading @2". Then the time parameters for using D11N in soil preparation and EX1100 for loading trucks are specified, and the project-specific atomic models can be generated. Finally the entire simulation model is assembled by automatically identifying different linking structures as required in the original sample application. In this case, a one-multiple indirect link is required between "soil preparation" and two "loading" *r-processes*, multiple-one direct link is required between two "loading" and one "hauling" *r-processes*, and one-one indirect link is required between "hauling" and "spreading" *r-processes*. This entire modeling process is schematically illustrated in Figure 4-41.

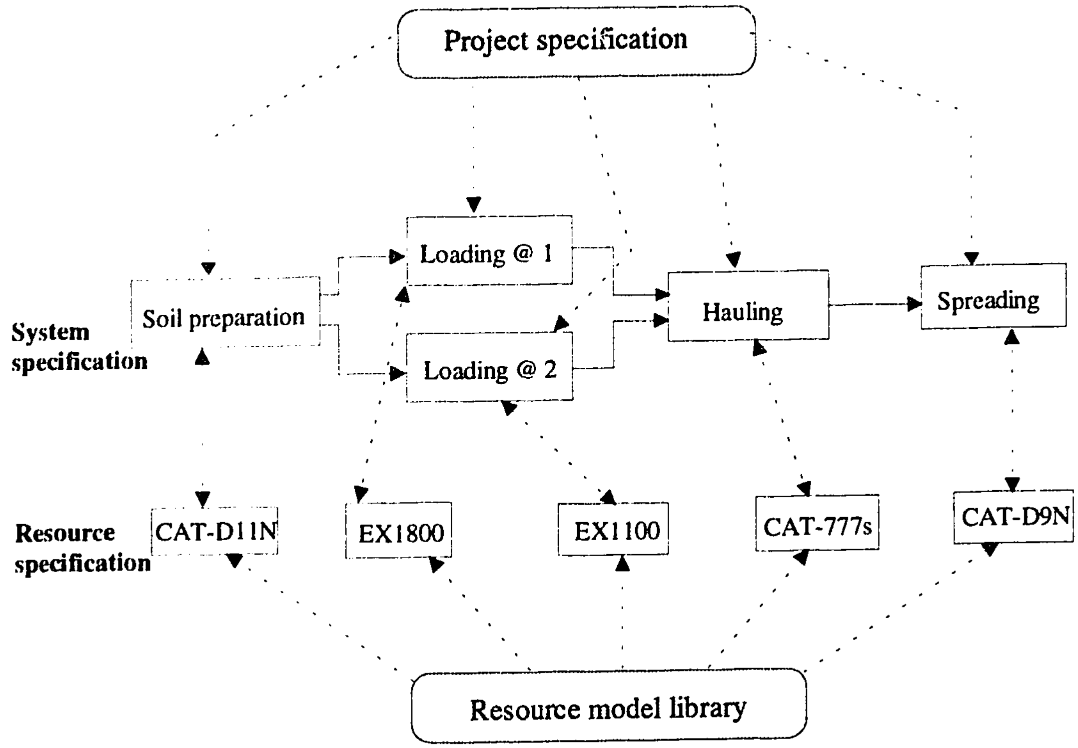


Figure 4- 41 The modeling process for the embellishment

By comparison, if a fixed process model library is used for the same situation, two identical process models must exist in the library. With the RBM, the same atomic library can be used to model different situations.

4.9 CONCLUSION

Traditionally, a project manager use deterministic methods in analyzing construction processes although real construction processes are stochastic, rather than deterministic. The production in a deterministic system increases linearly as the resources increases until a production ceiling is reached (Halpin and Woodhead, 1980). In a stochastic system, resources may wait for their matching resources to be available. In this sample application, the excavator may wait for a truck to be ready; also a truck may have to wait for the excavator. The production of a stochastic system will increase non-linearly as resources are increased. The increment rate will decrease gradually, especially close to the ceiling. Production estimates using simulation is closer to realistic situation than that using traditional deterministic methods.

A project manager may hesitate to construct a simulation model for this project because of the required level of simulation knowledge and the resultant time. However, the project manager can construct a simulation model for this project in little time (going through the resource and project specification process) by using this RBM environment without the requirement for proficiency in simulation. This approach is a cost-effective approach to assist the project manager in using computer simulation for analyzing and planning construction project.

The same technique can be applied to many other types of resource-dominated construction projects, or other industries in which resources dominates the work.

CHAPTER 5

AUTOMATED CONSTRUCTION SIMULATION OPTIMIZATION

5.1 INTRODUCTION

The resource-based modeling methodology presented in Chapters 3 and 4 can assist the user in building a simulation model for a construction project. After the simulation model is constructed, the user can experiment with it on a computer in an attempt to improve its performance. As described in Chapter 1, this process of improving a system's performance by using simulation is called simulation optimization (Pritsker 1986). Optimization is one of the most important aspects of construction simulation (Halpin et. al. 1989). Most construction operations in the real world are characterized by miscellaneous resources because they normally dominate project cost and duration. The objectives of construction optimization are usually to maximize production rate or minimize production cost by optimizing resource allocation. Therefore, the simulation optimization, in the context of this thesis, is defined as searching a feasible resource allocation to minimize or maximize the objective of a project. Researchers in construction simulation have attempted to achieve this goal by conducting sensitivity analysis.

Automated optimization techniques for construction simulation have not received enough attention from researchers. Traditional simulation-based optimization revolves around the comparison of exhaustive combinations of alternatives, most of which are manually driven. In complex systems, the number of combinations may be so large because of many variables that rendering the simulation almost be impossible (both the computer time and the comparison of outputs).

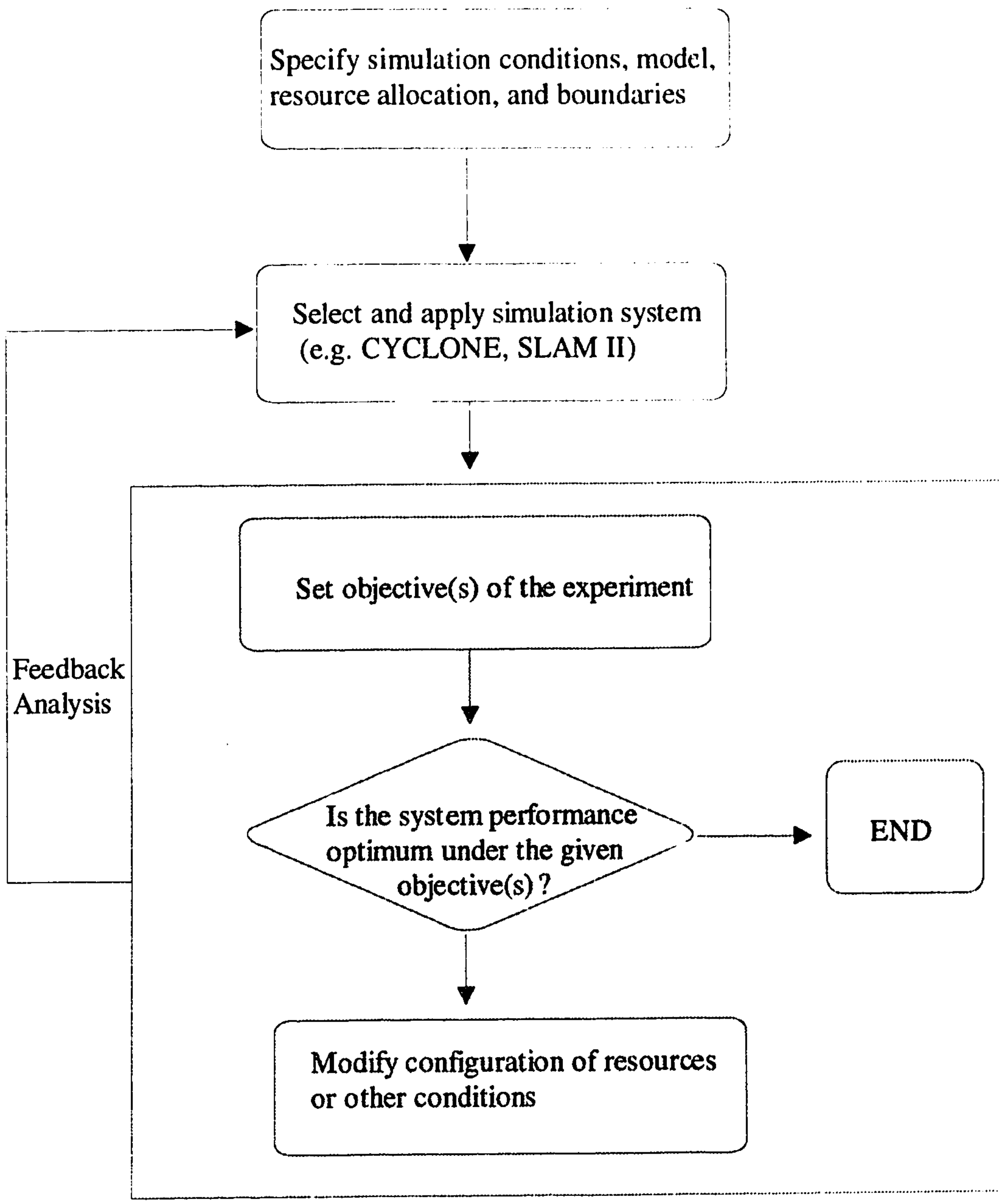


Figure 5- 1 An ideal simulation system

An ideal simulation system can be viewed as shown in Figure 5-1. In such a system the user provides initial resource input boundaries, and simulation conditions for a pilot run. The system would then experiment with various scenarios, analyze the performance and evaluate whether the current scheme is acceptable or not under the specified objectives. If performance can be improved, the system should automatically modify the resource configuration (or other conditions) to approach the desired objectives. In construction projects, a number of objectives can be identified. Among the most widely used are maximizing production rate, and optimizing resource utilization, and minimizing unit cost of the production.

This Chapter presents a technique that can supplement the heuristics applied in simulation feedback analysis. The advantage of this approach is its applicability to the automation of the simulation optimization procedure. By analyzing contributing resources and their queuing properties, algorithms related to three objectives are provided that can automatically drive the search for the best resource allocation that will optimize a particular objective.

In order to conduct simulation experimentation, a simulation engine which can run the simulation model has be used as the experimental media. Many general purpose simulation packages are available in the market including GPSS, SLAM II, ProModel, SIMAN, and SES/Workbench. CYCLONE, in the meantime, has been widely used in construction simulation for its simplicity. As noted in Chapter 1, the simulation engine is out of the scope of this research, and was adopted from the available resources. SLAM II, for its extensive modeling and simulation functions, was selected for the simulation language in Chapter 4 for implementing resource-based modeling methodology for earthmoving construction. However, in order to implement the automated optimization function proposed in this Chapter, it is necessary to access the code of the simulation engine. With the permission from Dr. Halpin at Purdue University, the developer of CYCLONE, we implemented the automated optimization approach in CYCLONE. However, the methodology is generic. It can be implemented in any simulation engine and can be used for any type of simulation system.

5.2 BASIC ELEMENTS IN PROCESS SIMULATION

The basic elements in construction process simulation systems are activities and queues. Activities represent the execution of the work tasks which use resources, consume time and money. Queues model the waiting states of resources or other entities in the system. Different simulation systems have different representation methodologies. In MicroCYCLONE, an activity can be simply represented by a square node while queue nodes are used to represent the waiting state of resources. Waiting has more flexible representation in SLAM II, such as QUEUE nodes, and AWAIT nodes. Resources can be initialized in queue nodes, resources blocks or INTLC statements.

In order to generalize the methods described in this Chapter we would first abstract the common features of simulation modeling as it applies to construction. All simulation languages can directly or indirectly provide statistics regarding the waiting of resources. MicroCYCLONE can provide resource waiting times at queue nodes in the form of average waiting times, and percent of time a queue node is occupied, for example. SLAM II provides waiting time and length at queue nodes, and resource utilization for resource blocks.

The resource *DELAY* statistic is defined as the fraction of time a resource is delayed at a service station (queue node). It can be calculated as follows:

$$DELAY = (\text{waiting time} / \text{total working time}) * 100 \qquad (5\text{-}1)$$

A resource may be used in several activities and wait at several locations, therefore, it may have more than one *DELAY* statistic. For convenience, the resources involved in a construction operation and their waiting locations can be tabulated as shown in Table 5-1.

Table 5- 1 *DELAY* statistics for Resources

	Queue position 1	2	...n
Resource 1	D_{11}	D_{12}	...n_1
Resource 2	D_{21}	D_{22}	...n_2
...	...	...	...
Resource m	D_{m1}	D_{m2}	...n_m

The *DELAY* statistic of a resource represents the degree of its usage in a system. If all resources have acceptable *DELAY* values, the performance of the system will normally be acceptable. By analyzing the *DELAY* statistics of participating resources in a simulation model, it can be determined whether the current resource allocation scheme is acceptable or not under a given objective. The values of this statistic will also enable the system to determine what resources should be changed in order to move closer to the "acceptable" allocation point.

From next section, three algorithms of construction simulation optimization are discussed in detail. They are: maximizing production rate, reasonable matching among resources, and minimizing unit cost separately.

5.3 MAXIMIZING PRODUCTION RATE

Production rate in the context of this Chapter is defined as the number of construction units produced per unit of time of operation. Alternatively it may be defined as the number of process-cycles completed per unit of time. This rate is often determined by the availability and numbers of participating resources (Berrios and Halpin 1988). It is limited by the capacity of the limiting resource (bottle-neck). The limiting resource may change depending on the allocation scheme of participating resources. In order to increase the production rate, the limiting resource must be manipulated. The limiting resource can be identified by using the *DELAY* statistic. If the *DELAY* statistics are represented as shown in Table 5-1, the limiting resource can then be identified as follows:

$$DELAY_{\text{(limiting resource)}} = \text{Min}_{\text{(all resources i)}} \{\text{Max}_{\text{(All waiting locations j)}} (D_{ij})\} \qquad (5\text{-}2)$$

$$DELAY_{\text{(surplus resource)}} = \text{Max}_{\text{(all resources i)}} \{\text{Min}_{\text{(All waiting locations j)}} (D_{ij})\} \qquad (5\text{-}3)$$

Equation (5-2) indicates that the resource with the least *DELAY* value will limit the value of the productivity for the process. When a resource is waiting at more than one location, the maximum *DELAY* encountered determines the productivity of the system (When that resource had the minimum *DELAY* associated with it). Opposite to limiting resource, the surplus resource can be defined by Equation (5-3), which means that the resource has enough capacity and will not restrict the system productivity under current configuration.

In order to illustrate how this can be used to automate the search for an allocation of resources that will maximize productivity, consider the CYCLONE model of an earth moving operation as shown in Figure 5-2. The process simply involves loading dirt at two locations. The loaders are served using a number of trucks.

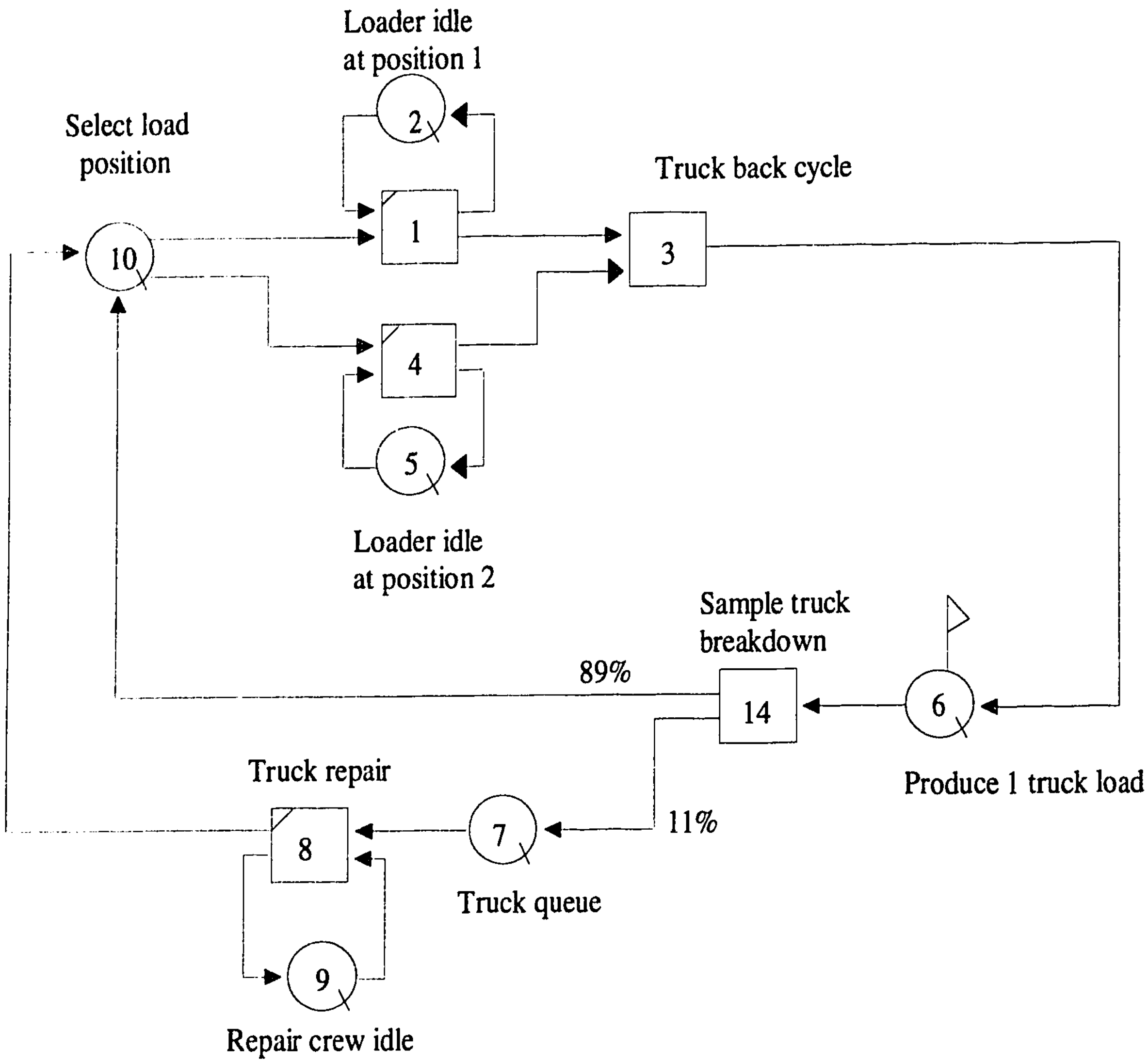


Figure 5- 2 CYCLONE model of an earthmoving operation

The simulation experiment was carried used MicroCYCLONE. The queue statistics collected from MicroCYCLONE are shown in Table 5-2.

Table 5- 2 Percent of time queues are occupied with 8 trucks in the system

Node #	2	5	7	9	10
Occupied time(%)	47.64	68.04	0.85	93.75	5.49

In this example, four kinds of resources can be identified. Trucks have two waiting positions at nodes 7 and 10 depending on whether they are queuing for loading or for repair. The *DELAY* statistics of the resources are given in Table 5-3.

Table 5- 3 *DELAY* values for resources

Resource	Location 1	Location 2
trucks	0.85	5.49
loader 1	47.64	
loader 2	68.04	
repair crew	93.75	

In order to increase the productivity of this system, the number of trucks should be increased because they have the lowest *DELAY* value.

In order to automate this procedure the following algorithm can be applied:

Define the initial numbers of the participating resources, resource queuing locations, and productivity improvement that will justify more iterations.

1. Conduct a pilot run to determine the current *DELAY* values.
2. Apply equation (5-2) to identify the limiting resource.
3. Increase the value of the limiting resource identified in Equation (5-2) if within specified boundaries.
4. Perform simulation experiment and determine the new production rate.

5. Repeat until the increment in productivity achieved is less than the specified value.

5.3.1 Sample implementation

This Appendix describes how the derived methods may be applied to automatically drive the process of resource allocation given that the a model of the construction operation has been developed using CYCLONE. The implementation was accomplished by appending the MicroCYCLONE sensitivity analysis module code described in (Halpin et al 1987). The discussion is limited to maximizing the production rate for brevity.

The user must first specify relevant parameters to the model. Model specific conditions such as required convergence conditions, queues where resources of interest are waiting, and initial numbers of resources, are samples of the requirements. The system will then conduct a pilot simulation run, apply the methods described in the paper and locate the best possible scenario of resource allocation. The algorithm driving the system may be summarized as follows:

1. Specify queue nodes where resource may wait for processing.

In order for automation of the simulation process to be feasible the essential resources that may contribute to production must be distinguished from general entities used to control the simulation experiment and other non-contributing resources. In CYCLONE models this is only possible if the user explicitly specifies which of the queue node contain contributing resources and which do not.

Only the resource numbers (capacity) can be directly changed at a queue node in CYCLONE. In addition, a resource might have more than one waiting location.

In order for the system to automatically locate the limiting resource (bottle-neck) as described in (Bernold 1985), the user must specify and identify resources and their correspondent waiting locations.

2. Specify convergence conditions (minimum improvement on production rate)

In general if the limiting resource is increased, production rate may increase until another resource becomes the limiting resource. Specifying the convergence limits allows the system to terminate the operation once the improvement in the production rate is less than a desirable limit.

3. Set initial resource configuration for pilot simulation run

The system interfaces with MicroCYCLONE to perform the simulation experiments and as such requires an initial CYCLONE model with initial data sets to perform the simulation experiments.

Let $P'=0$ (initial production rate)

4. Conduct a simulation run

Given a current resource configuration, the system calls the main simulation module of MicroCYCLONE to simulate the system for a given cycles or length.

Let P={production rate of this run}

5. Compare production rate between last two runs

If $P - P' \leq$ {given minimum improvement}then go to step 8. Otherwise continue.

6. Compare *DELAY* values and identify the limiting resource

After a simulation run, simulation results as shown in Table 1 are obtained. Equation 2 is then used to identify the limiting resource. If the current capacity of this resource reaches its upper boundary, Equation 2 is used again to find the next limiting resource, i.e.,

$$X_i < \{\text{upper boundary of resource } i\}$$

X_i----current capacity of resource i.

7. Increase the value of the limiting resource by δ, i.e.

$$\text{Let } X_i = X_i + \delta$$

δ ---- resource increment (1 or any given value.)

go to step 4.

8. End

5.3.2 A Sample Application

In order to demonstrate the practicality of this method it has been implemented in MicroCYCLONE. The Sensitivity Analysis (Halpin et. al 1989) module of the program was found to be a suitable medium for the implementation as it provides the necessary user interface.

The model shown in Figure 5-2 was considered for optimization. It was decided that an added value of 2% on the previous hourly production rate will be the minimum amount that will justify further resource manipulation. The system starts with the initial allocation of 10 trucks at node 10 and 1 loader each at nodes 1 and 5. The user is presented with the menu where the first option to maximize productivity is selected.

The program will then list the initial resource allocation in the model and request that the user identifies the resources that will participate in the analysis and their possible waiting locations. The minimum improvement of 2% on productivity is also specified by the user as shown in Figure 5-3.

RESOURCE CHANGES				
Q. Label	Description	No. of Res.	V-COST	F-COST
2	EXCAVATOR IDLE AT POSITION 1	1	0.00	0.00
5	EXCAVATOR IDLE AT POSITION 2	1	0.00	0.00
7	TRUCK QUEUE	0	0.00	0.00
9	REPAIR CREW IDLE	1	0.00	0.00
10	SELECT LOADING POSITION	10	0.00	0.00

... end of list ...

What differential increment in productivity justifies termination of the operation: **0.02**

Figure 5- 3 Sample query screen from MicroCYCLONE

The simulation will then be automatically driven by the algorithm outlined in the previous section and the results reported upon completion of the process as shown in Table 5-4.

Table 5- 4 Reports from the automated resource allocation optimization

Run number (1)	Total time (2)	Production/hour (3)	Q-label (4)	Resource (5)
1	69.95	21.44	10	11
2	64.35	23.31	10	12
3	59.82	25.07	2	2
4	57.14	26.25	10	13
5	51.99	28.85	10	14
6	50.62	29.63	10	15

The results show that the maximum production rate can be attained when 15 trucks are allocated to node 10, two loaders at node 2 with all other nodes unaffected. It should be mentioned that the 2% minimum increment on production rate was an arbitrary choice for this example.

5.4 REASONABLE MATCHING AMONG RESOURCES

In a stochastic system, it is difficult to keep all resources at full utilization. When the utilization of one resource is increased, the utilization of its matching resource(s) will decrease. Because the operating cost of resources is different, it is desirable that resources with higher operating costs are kept at higher utilization. In order to run a system economically, waiting rates among resources should be kept at reasonable ratios. In order to obtain reasonable matching among resources, one can either increase 'limiting' resources or decrease 'surplus' resources. Whether an increment or decrement is reasonable can be evaluated from the *DELAY* cost of participating resources. The

DELAY cost is the system operating cost while non-productive (i.e. waiting for processing) as defined by Equation (5-4).

$$C_w = \sum_{i=1}^{m} c_i \sum_{j=1}^{n_i} D_{ij} l_{ij} \qquad (5\text{-}4)$$

Where:

C_W = total *DELAY* cost of the system ($/unit of time, e.g. $/hr)

c_i = cost per unit of time of resource i ($/hr)

l_{ij} = the expected queuing length of resource i at position j

m = number of resources

n_i = number of waiting positions of resource i

When the capacity of resource i is changed, the *DELAY* cost will change to C_W':

$$C'_w = \sum_{i=1}^{m} c_i \sum_{j=1}^{n_i} D'_{ij} l'_{ij} \qquad (5\text{-}5)$$

It is desirable to decrease the *DELAY* cost. Therefore C_W' should be less than C_W or the change $\Delta C_W = C_W - C_W'$ defined in Equation (5-6) would be positive in value.

$$\Delta C_w = C_w - C'_w = \sum_{i=1}^{m} c_i \sum_{j=1}^{n_i} (D_{ij} l_{ij} - D'_{ij} l'_{ij}) \qquad (5\text{-}6)$$

An algorithm for automating resource allocation based on the reasonable matching among resources can be established as follows:

1. Given the system boundaries, initial resource allocation and hourly cost.
2. Conduct a pilot simulation run.
3. Apply Equation (5-2) to determine the candidate resource for increase.

4. Conduct another simulation run with new resource allocation, and evaluate the parameters given in Equation (5-6).

5. If $\Delta C_W > 0$ then keep moving in the same direction, otherwise apply Equation 3 to decrease the surplus resource, and go to step 4.

6. If $\Delta C_W <= 0$ for two directions or resources reached their boundaries, then stop procedure.

5.5 MINIMIZING THE UNIT COST

Minimizing the cost per unit of production is a desirable objective on a practical level in construction as it determines the entire cost of the operation. The unit cost of production can be manipulated by either increasing hourly production rates or decreasing the system's operating cost. The hourly cost (or cost per unit of time) of an entire construction process can be defined as given in Equation (5-7):

$$C = \sum_{i=1}^{m} r_i c_i \qquad (5\text{-}7)$$

Where

C = total cost per unit of time for the entire construction process (e.g. \$/hr).

r_i = number of units of resource i

c_i = cost per unit of time of resource i (e.g. \$/hr)

The unit cost will then be defined as given in Equation (8):

$$U = C / P \qquad (5\text{-}8)$$

Where,

U = Cost per unit of production (e.g. \$/unit)

P = Production rate (e.g. units produced /hr)

When the capacity of resource i is increased by an increment Δr_i, the production rate will change to P' where:

$$P' = P + \Delta P \qquad (5\text{-}9)$$

the total operating cost of the system will be C' where:

$$\begin{aligned} C' &= c_1 r_1 + \ldots + c_i(r_i + \Delta r_i) + \ldots + c_m r_m \\ &= C + \Delta r_i c_i \end{aligned} \qquad (5\text{-}10)$$

The unit cost will therefore be represented by:

$$U' = C' / P' = \frac{C + \Delta r_i c_i}{P + \Delta P} \qquad (5\text{-}11)$$

The increment in unit cost is given as:

$$\Delta U = U' - U = \frac{C + \Delta r_i c_i}{P + \Delta P} - \frac{C}{P} = \frac{\Delta r_i c_i P - C \Delta P}{P(P + \Delta P)} \qquad (5\text{-}12)$$

In order to decrease the unit cost, then

$$\Delta U < 0 \quad \Rightarrow \quad \Delta r_i c_i Q - C \Delta Q < 0$$

or

$$\Delta P > \frac{\Delta r_i c_i P}{C} \quad \Rightarrow \quad \frac{\Delta P}{P} > \frac{\Delta r_i c_i}{C} \qquad (5\text{-}13)$$

An algorithm for automating resource allocation based on minimizing unit cost can then be established as follows:

1. Given the system boundaries, initial resource allocation and hourly costs.
2. Conduct a pilot simulation run.
3. Apply Equation (5-2) to determine the candidate resource for increase.

4. Conduct another simulation run with new resource allocation, and evaluate the parameter given in Equations (5-7) and (5-8).

5. Apply Equations (5-12) and (5-13).

6. If $\frac{\Delta P}{P} > \frac{\Delta r_i c_i}{C}$ then keep moving in the same direction of resource increment otherwise apply Equation (5-3) to determine the candidate resource decrease, and go to step 4.

7. If $\frac{\Delta P}{P} \leq \frac{\Delta r_i c_i}{C}$ for two directions or resources reached their boundaries then exit.

The same earth-moving operation modeled in Figure 5-2 is used to illustrate the working of minimizing unit cost for a construction process. With eight (8) trucks allocated at node 10, one loader at node 2 and node 5 the production rate was found to be 246 units/hr at a cost of $694 per hour yielding a unit cost of $2.82 per unit.

The results are scanned according to the algorithm presented above and the lowest *DELAY* value is found to be associated with the trucks. Trucks are increased by two units yielding the following new results, C' = $795/hour, P'=308 units/hour, and U'=$2.58/unit. By applying Equations 12 and 13, $\Delta P/P > \Delta r_i c_i/C$ can be assessed and as a result the number of trucks should be increases further. The remainder of the iterative procedure is given in Table 5-5.

Table 5- 5 Results of Minimum Unit Cost

# Trucks	8	10	12	14	16	18	20
Waiting at QUE 2 (%)	47.64	38.92	30.40	23.18	15.81	9.12	3.1
Waiting at QUE 5 (%)	68.04	56.16	44.37	32.07	20.95	11.25	3.6
Waiting at QUE 7 (%)	0.85	1.73	2.74	4.02	6.06	8.17	10.2
Waiting at QUE 9 (%)	93.75	90.47	86.57	82.32	78.14	73.99	68.73
Waiting at QUE 10 (%)	5.49	10.87	17.13	27.12	40.50	58.85	79.25
P	246	308	367	425	479	526	564
C	694	795	897	999	1101	1202	1304
U	2.8220	2.580	2.445	2.351	2.300	2.284	2.311
ΔP	N/A	62	59	58	54	47	38
$\Delta r_i c_i$	N/A	101	101	101	101	101	101
$\Delta P/P$	N/A	0.252	0.192	0.158	0.147	0.098	0.072
$\Delta r_i c_i/C$	N/A	0.146	0.127	0.113	0.101	0.092	0.084
$\Delta P/P >? \Delta r_i c_i/C$	N/A	yes	yes	yes	yes	yes	no

The minimum unit cost attainable on the operation is associated with 18 trucks. Further increase in number of trucks will increase the unit cost.

5.6 CONCLUSION

Automated modeling presented in Chapters 3 and 4 and the automated optimization presented in this Chapter represent two important steps towards the full automation of a simulation process. With the automated optimization function, the user can directly locate the optimum resource allocation. It can save time for the modeler by avoiding many unnecessary simulation combinations.

This heuristic automated optimization approach is generic. It can be implemented in any simulation language to optimize resource allocation for any construction project.

The three objectives can only be manipulated independently. Another potential research topic in future work is how to consider these multiple objectives in one system to satisfy practitioner's requirement.

CHAPTER 6

AN OPTIMIZATION METHOD FOR SIMULATING LARGE COMPLEX SYSTEMS

6.1 INTRODUCTION

One of the major purposes of analyzing a real world system is to optimize its performance which might result in improvement in production or cost saving. Traditionally, various analytic techniques have been widely developed and used in analyzing and optimizing system's operations because analytic models can be easily constructed and be readily solved. Simulation has been categorized as a technique of last resort since it tends to be expensive (Grain et. al 1992). With the advancement of simulation techniques, simulation is becoming faster and cheaper. Moreover, simulation can represent a real world problem at a level of detail that is usually beyond the scope of analytic models.

To avoid an exhaustive enumeration, Chapter 5 presented a heuristic method which can automatically search for the "optimum" resource allocation for specified objectives. However, for a large system with many decision variables or these variables with a large number of states, the required simulation combinations tend to be large and impractical. The experimentation would consume too much computer time. Moreover, when a general simulation language is used to model a large and complex system, it is usually difficult to model the constraints amongst its components. Another problem with current microcomputer-based special purpose simulation languages (e.g. CYCLONE) is that their applicable scope is limited to small or medium size models.

As an extension to the automated optimization method presented in Chapter 5, a hybrid method that incorporates the advantages of simulation and analytic techniques is proposed in this Chapter to optimize the performance of a system composed of multiple queuing processes. Such systems are often characterized by randomness and complexity within each individual process, and logical or physical constraints define the interactions between the various processes.

This method consists of two steps in optimizing a real world system: first, to independently simulate the operation of each process at selected states; then, a mathematical model based on the constraints among these processes is constructed while incorporating the simulation results of all processes. The optimal solution of the system is located analytically by solving the mathematical model.

6.2 BREAKDOWN OF LARGE SYSTEMS

A large complex system can be broken down into basic components (e.g. processes for an operational system), which operate independently and can be linked together through input-output ports. If a process is represented by a box, then these boxes form a series-parallel system as shown in Figure 6-1.

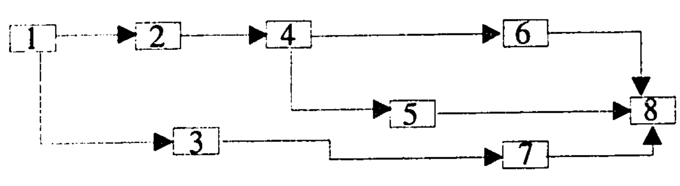


Figure 6- 1 A sample system

Each process has a set of decision variables ($x_i=\{x_{i1}, x_{i2}, ...\}$) that determine its operating state. In construction engineering, x_i can represent a system resource, and is discrete. For example, the number of trucks in the hauling process of an earthmoving construction system is a decision variable, which determines the states of this system. In discrete simulation (many engineering applications), the decision variables are discrete.

Let S_i represent the state variable of process i, numbered $\{1, 2, ..., s_i\}$. Where s_i is the total number of states of processes i, determined by the range of decision variables. Considering the same earthmoving example as above, and assuming that the number of trucks can change from four to eight, then five states of the process can be defined as 1 to 5 where state 1 represents four trucks, state 2 represents five trucks, and so on.

A simulation model for each process can be constructed and simulation experiments may be conducted at the possible states. The simulation results estimating the production and cost of the process at each state can be summarized in a matrix form as shown in matrix (6-1).

$$\begin{bmatrix} 1 & 2 & \cdots\cdots & s_i \\ x_{i1} & x_{i2} & \cdots\cdots & x_{is_i} \\ c_{i1} & c_{i2} & \cdots\cdots & c_{is_i} \\ p_{i1} & p_{i2} & \cdots\cdots & p_{is_i} \\ u_{i1} & u_{i2} & \cdots\cdots & u_{is_i} \end{bmatrix} \quad \text{For process} \quad i = 1,2,\ldots\ldots,n \qquad (6\text{-}1)$$

$$u_{ij} = c_{ij} / p_{ij} \qquad \text{For state} \quad j = 1,2,\ldots\ldots,S_i$$

Where n represents the total number of processes in a system; x_{ij} is the state variable; c_{ij} and p_{ij} represent the operating cost and productivity of process i at state j separately; u_{ij} is the unit cost.

Constructing simulation models and experimenting with them have been discussed in many reference (Pritsker 1986 and Halpin 1990), and as such will not be addressed in this Chapter. The objective of this Chapter is to discuss how to construct the hybrid mathematical model and how to solve it.

6.3 MATHEMATICAL MODELS AND THEIR SOLUTIONS

For the purpose of this work, a system can be classified as 1) parallel, 2) series, or 3) mixed parallel-series. The mathematical models for the three types of systems are discussed separately from next section.

6.3.1 Parallel Processes

A parallel system is defined as one in which the outputs of all processes are linked to the same port as shown in Figure 6-2. For an independent process, the state with the minimal unit cost is usually the optimum state (local optimum). If no constraints exist among these processes, they can be simulated independently, and the combination of the local optimum states of all processes forms the overall optimum solution of the entire system.

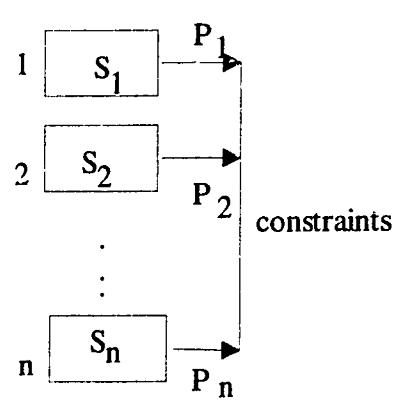


Figure 6- 2 Parallel system

However, in many real world systems, constraints usually exist among these parallel processes. An example of a common constraint is: the output quantities amongst processes should maintain fixed ratios. Consider for example, a concrete precasting plant that consists of three production lines. Line one produces UNIT I, line 2 produces UNIT II, and line 3 packages these units into one package which is composed of one UNIT I and two UNIT IIs. This system is shown in Figure 6-3.

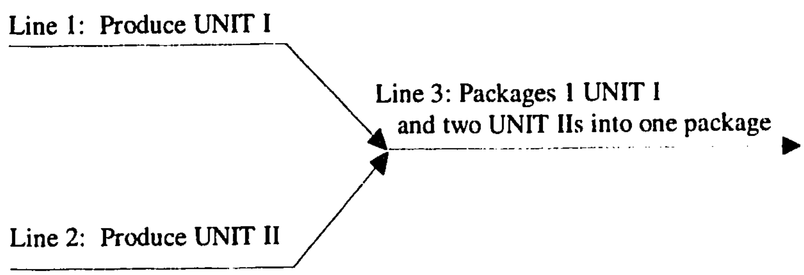


Figure 6- 3 A precasting example

Generalizing this concept, the constraints of parallel processes can be expresses as:

$$P_1/P_2=\lambda_1$$

$$P_2/P_3=\lambda_2$$

$$\cdot$$

$$\cdot \qquad (6\text{-}2)$$

$$\cdot$$

$$P_{n-1}/P_n=\lambda_{n-1}$$

Where P_1, P_2, ..., P_n represent the output (e.g. production) quantities of process 1, 2,......, n; and λ_1, λ_2, ..., λ_{n-1} are constants that represent the required ratios between two related processes.

(1) The optimum state of parallel system

The system satisfying the constraints in Equation (6-2) is referred to as balanced system, and the corresponding state is defined as a balanced state of the system. The optimum state of a parallel system can only take place at a balanced state. Further, the outputs of processes at balanced states are defined as balanced outputs.

There are n variables and n-1 equations in Equation (6-2). By definition, $P_1, P_2, \ldots, P_n$ are the balanced outputs of processes 1, 2, ..., n. In order to solve for $P_1, P_2, \ldots, P_n$, we must have one more equation or know the value of one of the $P_1, P_2, \ldots, P_n$. In order to locate a balanced state of the system, an arbitrary process is defined as the reference and is labeled as process 1 to initiate the search. Based on the output of the reference as shown in (6-1) and required ratios defined in equation (6-2), the balanced outputs of the other processes can be calculated. Then the balanced output of each process can be used to determine its corresponding decision variable or state.

Let S be the sate number of the reference process, and P_{ij} represent the balanced output of process i at reference state j. For the reference process,

$$P_{1j} = p_{1j} \qquad j=1, 2, \ldots, S \tag{6-3}$$

The balanced outputs for other processes can be computed using (6-2):

$$P_{ij} = P_{i-1j} / \lambda_{i-1} = P_{1j} / (\lambda_1 \lambda_2 \ldots \lambda_{n-1}) \tag{6-4}$$

For discrete event simulation, the real output values obtained from the simulation experiment may not exactly equal the values required by Equation (6-4). In order that this process does not impede the overall performance of the system, the real output quantity should be equal or greater than the balanced quantity calculated by Equation (6-4). On the other hand, increasing the output of a process will increase the operating cost of the process. Therefore, the state with the minimal output that can satisfy Equation (6-4) should be the balanced state of process i as follow:

$$k = \min_{1 \le l \le S_i} \{p_{ik} \mid p_{il} \ge P_{ij}\} \tag{6-5}$$

When the real output is greater than the balanced output, the surplus amount will not contribute to the overall production. Therefore, the expected output of this process should be equal to the balanced output instead of the real output. However, the operating cost of a process determined by its decision variables (or operating state) should correspond to the real output state.

Let C_{ij} represent the operating cost of process i at balanced state j, then

$$C_{ij} = c_{ik} \tag{6-6}$$

Summarizing all balanced states, we get:

$$\begin{array}{ccccc} \text{Process} & 1 & 2 & \dots & S(\text{state}) \\ 1 & P_{11}/C_{11} & P_{12}/C_{12} & \dots & P_{1S}/C_{1s} \\ 2 & P_{21}/C_{21} & P_{22}/C_{22} & \dots & P_{2s}/C_{2s} \\ . & \dots & \dots & \dots & \dots \\ n & P_{n1}/C_{n1} & P_{n2}/C_{n2} & \dots & P_{ns}/C_{ns} \end{array} \tag{6-7}$$

Conclusion 1: The state j* which satisfies Equation (6-8) will be the optimal state of a parallel system.

$$j^* = \min_j \{\sum_{i=1}^{N} C_{ij} / \sum_{i=1}^{N} P_{ij}\} \tag{6-8}$$

This conclusion can be proved as follows. According to Equation (6-2), the total product of the system at state j is

$$\sum_{i=1}^{N} Q_{ij} = P_{1j} + P_{2j} + \ldots + P_{Nj}$$
$$= P_{1j} + P_{1j} / \lambda_1 + P_{1j} / (\lambda_1 \lambda_2) + \ldots + P_{1j} / (\lambda_1 \lambda_2 \ldots \lambda_{N-1}) \qquad (6\text{-}9)$$
$$= P_{1j}[1 + 1 / \lambda_1 + 1 / (\lambda_1 \lambda_2) + \ldots + 1 / (\lambda_1 \lambda_2 \ldots \lambda_{N-1})]$$

because $\lambda_1, \lambda_2, \ldots, \lambda_{N-1}$ are constants, then

$$\lambda = 1 + 1/\lambda_1 + 1/(\lambda_1 \lambda_2) + \ldots + 1/(\lambda_1 \ldots \lambda_{N-1}) \qquad (6\text{-}10)$$

is a constant; therefore right side of equation (8) can be expressed as:

$$\min_j \{\sum_{i=1}^{N} C_{ij} / \sum_{i=1}^{N} P_{ij}\} = \min_j \{\sum_{i=1}^{N} C_{ij} / (\lambda P_{1j})\} = \lambda \min_j \{\sum_{i=1}^{N} C_{ij} / P_{1j}\} \qquad (6\text{-}11)$$

From a project management perspective, the goal is to complete a given amount of output (production) with the minimum cost. Since the matching relationships among processes are fixed, the total output of the reference process can determine the total output of the entire system. Let the target production (output) of reference process be P. The required operation period of this system in order to fulfill the target production is:

$$t_j = P/P_{1j} \qquad (6\text{-}12)$$

Obviously each process should work the same periods. The total operating cost of the system (at state j) is:

$$C_j = \sum_{i=1}^{n} t_j C_{ij} = \sum_{i=1}^{n} PC_{ij} / P_{1j} \qquad (6\text{-}13)$$

The objective is to minimize the total operation cost of the system, i.e.,

$$\min_j \{\sum P.C_{ij} / P_{1j}\} = P.\min_j \{\sum_{i=1}^{n} C_{ij} / P_{1j}\} \qquad (6\text{-}14)$$

P in Equation (6-14) is a constant. Minimizing parts of equations (6-11) and (6-14) are the same; and equation (6-11) is another expression of conclusion 1. Therefore, conclusion 1 holds.

(2) Simplifying A Compound Process

Processes may be conventionally divided into two types: simple and compound. A simple process possesses one output port; otherwise the process is compound. In a real world system, some processes may have more than one output port. Also, several simple processes may be combined to form a compound process. A compound process must be simplified into a simple process before it can be integrated with other simple processes. One of the following methods can be used to accomplish this:

a) Decomposition method

Compound processes in a parallel system, if possible, should be decomposed into simple parallel processes. If there are c outputs in the ith process, this process can be decomposed into c simple parallel processes that can be added to the original system. The constraints among the c processes and their operating cost can be determined by the operating features of this original process. For example, the screening process of an aggregate plant can produce two products, and can be decomposed into two processes where each produces one product.

b) Summation method

If all outputs in a compound process are physically the same and are additive, then the addition of all outputs in the process can be taken as the output of the process, and the compound process may then be treated as a simple process. For the same screening example, we can simply add the two products into one.

c) Simple replacement

If a compound process cannot be simplified by one of the above two methods, and there is only one output in this process that is related to other processes, this related output can be taken as the replacement of this process output, but the operating cost should cover the entire process. Still considering the same screening example, if one of the two products is a by-product and is useless, then it is reasonable to take the useful product to represent the output of the process.

d) Partial replacement

If some (or part of the) outputs of a compound process are related to other processes, these partial outputs can be taken out as the replacement of this process, then it can be treated by decomposing it or using the summation method. Also the operating cost should cover the entire process.

(3) Equivalent process

An equivalent process is defined as the combination of several simple processes. The output of an equivalent process represents the fundamental features of integrated processes, and can be computed by choosing one of the following rules according to the physical features of a system:

Rule 1: If the outputs of all processes are physically the same, and their quantities are additive, the equivalent output of parallel processes equals the sum of the outputs of all processes.

Rule 2: If the maximum output is the major concern of the system, the equivalent output of the parallel processes equals the maximum output among all processes.

Rule 3: If the minimal output is the major concern of the system, the equivalent output of parallel processes equals the minimum output among all processes.

Rule 4: If the combination of the process's outputs produces a new output, the equivalent output of parallel processes should be determined by a transfer function.

6.3.2 Series Systems

A series system can be defined as one where two or more processes are serially linked through input/output ports with one process not starting until its predecessor is complete. Also each process has only one input port and one output port. A general series system is shown in Figure 6-4. Where I_i and P_i represent the input and output of process i. S_i is the state number of process i. n is the number of series processes.

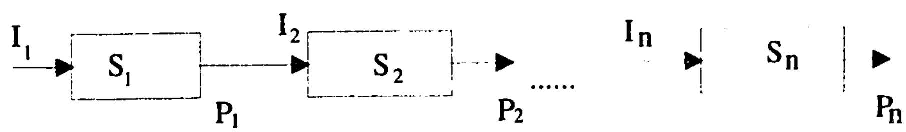


Figure 6- 4 The series system

Let P_1, P_2, ..., P_n and I_1, I_2, ..., I_n represent the real outputs and inputs of processes 1, 2, ..., n, and assume $P_i = I_{i+1}$ holding for all processes except process k ($P_k \neq I_{k+1}$). Let's assume an inventory between processes k and k+1, and the inventory quantity is W_k, then

$$P_k = I_{k+1} + W_{k+1}$$

Two cases can occur including $W_{k+1} > 0$ and $W'_{k+1} < 0$, Case 1: If $W_k > 0$, then surplus quantity of inventory occurs. That means the system will put W_{k+1} amount of output from its previous process into inventory in each unit of time. If the system keeps operating, inventory will increase with the working time.

Case 2: If $W_k < 0$, then a shortage in inventory occurs. That means the processes from k to n have surplus production capacity since there is not enough inputs.

In conclusion, in order to optimize the overall performance of a series system, the inventory quantity between any two series processes should be zero, i.e. only at balanced state.

Conclusion 2: For a series system operating at steady state, if the states at which the expected output from the previous process equals the expected input to its following process in a unit of time are defined as the balanced states, then the system can reach its optimum only at a balanced state.

Conclusion 2 can be expressed as:

$$I_i = P_{i-1} \qquad i = 1, 2, ..., n \qquad (6\text{-}15)$$

Assume the transfer function between input and output of the ith process as:

$$P_i = T_i(I_i) = T_i(P_{i-1}) \qquad (6\text{-}16)$$

The transfer function of each process can be determined from its operating feature. Beginning from the output of the first process, step by step, the outputs of the following processes can be derived. Generally speaking, the last output in a series system is the expected output, and the outputs of all other processes are the transfer results.

Rule 5: The equivalent output of series processes is equal to the balanced output of the last process in the series.

If the first process in this series group is selected as the reference, according to (6-1) and rule 5, beginning from an output P_{1j} of the reference process, the corresponding equivalent outputs P_{nj} and operating costs of the intermediate processes can be calculated. Adding up the operating costs, the output and corresponding operating cost of the equivalent process of a series group can be summarized as in (6-17),

$$\begin{matrix} P_{n1} & P_{n2} & \ldots & P_{nS} \\ \sum_{i=1}^{n} C_{i1} & \sum_{i=1}^{n} C_{i2} & \ldots & \sum_{i=1}^{n} C_{iS} \end{matrix} \qquad (6\text{-}17)$$

The objective is to minimize the cost per unit output.

Conclusion 3: The state j^* determined from equation (6-18) is the optimum operating state of a series system.

$$j^* = \min_{j} \{ \sum_{i=1}^{n} C_{ij} / P_{nj} \} \qquad (6\text{-}18)$$

If any process is compound in the series system, some steps should be taken to simplify it into a simple process as discussed in the previous section.

6.3.3 Mixed Parallel-Series System

A mixed system (parallel-series) can be treated like a parallel-series electric circuit. At first, pure parallel process groups or series process groups are merged into the correspondent equivalent processes by using the methods detailed in previous sections. An equivalent process is then simplified into a simple process; repeat above steps until the entire system is merged into one equivalent process. Like solving a dynamic programming model, the optimization procedure is divided into two stages: forward and backward. For each equivalent process, the balanced states associated with balanced outputs and costs are obtained. After the entire system is merged into one equivalent process, the optimum state can be selected based on the minimum unit production cost. Tracing backwards from this point, the optimum state of each equivalent or simple process is to be determined that leads to the overall optimum.

6.4 A SAMPLE APPLICATION

An earthmoving project is composed of three cut locations and one fill location. The three cut locations provide three types of materials: clay, sand, and rock which will be used in the fill area. There are three adjacent zones to be filled with each of the materials. The design requires that the ratios for the three materials are 1:0.85:3.0, i.e. one unit of clay, 0.85 unit of sand, and three units of rock. Three hauling processes are used to move each type of material separately. The material is dumped in the required zone in the proper proportions and then compacted. The clay zone has its own compactors; the sand and rock zones share the same compactors. The system is illustrated in Figure 6-5. The objective is to optimize the equipment allocation in order to minimize the overall construction cost. The available equipment for the project and their operating parameters are given in Table 6-1

Table 6- 1 Available equipment and their operating parameters

	Loaders			Trucks			Compactors		
	Number	Loading time (min.)	Cost rate ($/h)	Number (min-max)	Capacity (m^3)	Cost rate ($/h)	Number (min-max)	Compacting time (min.)	Cost rate ($/h)
Clay	1	N(3,1)*	130	2 - 8	10.5	51	1 - 6	N(20,10)	324
Sand	1	N(3,1)	130	2 - 10	10.5	51	4 - 14	N(15,5)	324
Rock	2	N(5,3)	223	7 - 20	15.8	81		N(20,5)	

* Note: N(3,1) means the loading time is a Normal distribution with mean 3 and standard deviation of 1.

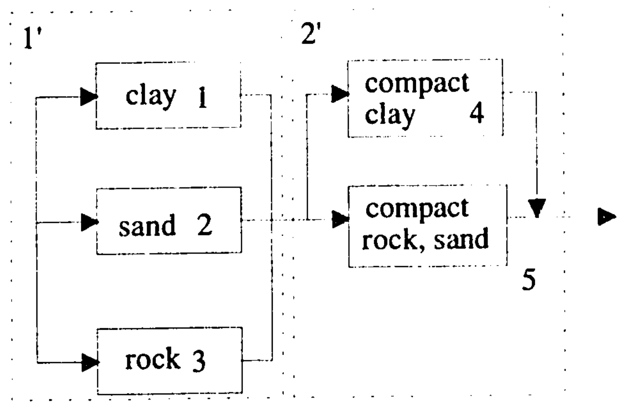


Figure 6- 5 An earthmoving example

CYCLONE can not directly model the entire system in one simulation model, because: 1) it can not represent the production constraints required for the three materials; 2) it allows only one production COUNTER node in a model. However, it is easy to use CYCLONE to model and simulate the five processes independently.

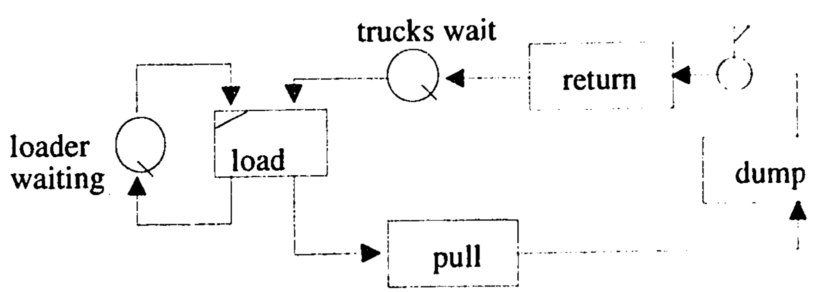


Figure 6- 6 Loading and hauling process

The three hauling processes can share the same CYCLONE-based model as shown in Figure 6-6. For different processes, correspondent parameters should be given, including lo' 'er number, cost rates, loading time, truck number, hauling time, dumping time and return time.

Simulation experiments were conducted based on the model shown in Figure 6-6 and using the parameters required for each process (i.e. cost, trucks, etc.). The simulation results using CYCLONE for the three hauling processes are given in Tables 6-2, 6-3, and 6-4.

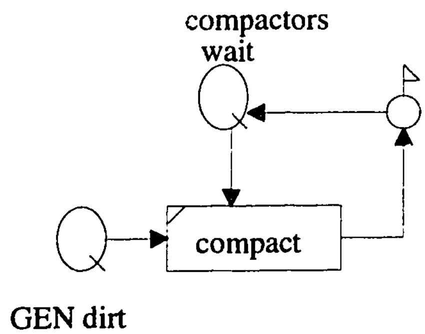


Figure 6- 7 Compacting process

The compaction process of clay can be modeled as shown in Figure 6-7. The simulation experiment is carried in a manner similar to the hauling process, and the results are shown in Table 6-5.

The compaction process of sand and rock is a compound process which produces two outputs. Because the compaction production for sand and for rock is additive, the summation method can be used to simplify it into a simple process. Since the design requires a 0.85:3 ratio between sand and rock, the compaction equipment should produce its output according to this ratio. The model shown in Figure 6-7 is used to simulate the compaction process of sand and rock separately. These results are then combined into one table.

If cost, productivity and unit cost for compacting sand and rock are represented as C_s, P_s, u_s, and C_r, P_r, u_r respectively, the outputs of the simple process as shown in Table 6 are calculated using the following formulae.

$$C = (0.85*C_s+3*C_r)/3.85 = C_s = C_r$$

$$P = (0.85*P_s+3*P_r)/3.85$$

$$d = (0.85*u_s+3*u_r)/3.85$$

Where the coefficients 0.85 and 3 are the required ratios for sand and rock and each formula is divided by 3.85 to standardize the cost, production rate, and unit cost. m is the number of equipment; C is cost per hour; P is the productivity per hour; u is the unit cost.

Table 6- 2 The hauling process of clay (process 1)

m	2	3	4	5	6	7	8
C	232	283	334	385	436	487	538
P	30.43	44.21	60.60	75.27	87.13	100.87	115.71
u	7.62	6 40	5.51	5.12	5.00	4.83	4.65

Table 6- 3 The hauling process of sand (process 2)

m	2	3	4	5	6	7	8	9	10
C	232	283	334	385	436	487	38	589	640
P	21.70	31.46	43.53	54.33	62.81	73.08	85.20	89.41	103.04
u	10.69	8.99	7.67	7.09	6.94	6.66	6.31	6.59	6.21

Table 6- 4 The hauling process of rock (process 3)

m	7	3	9	10	11	12	13	14	15	16	17	18	19	20
C	1013	1094	1175	1256	1337	1418	1499	1581	1661	1742	1823	1904	1985	2070
P	125	144	153	174	182	190	204	225	234	245	253	264	270	275
u	8.13	7.59	7.68	7.21	7.35	7.45	7.34	7.01	7.09	7.11	7.21	7.22	7.346	7.50

Table 6- 5 The compacting process of clay (process 4)

m	1	2	3	4	5	6
C	324	648	972	1296	1620	1944
P	30.84	61.23	91.78	118.89	148.51	177.90
u	10.51	10.58	10.59	10.90	10.91	10.91

Table 6- 6 The compacting process of sand and rock (process 5)

m	4	5	6	7	8	9	10	11	12	13	14
C	1296	1620	1944	2268	2592	2916	3240	3564	3888	4212	4536
P	164	207	244	282	320	349	383	415	443	474	503
u	6.74	6.68	6.80	6.88	6.92	7.15	7.22	7.34	7.51	7.59	7.71

Solution: The processes 1, 2, 3 constitute a parallel process group and can be merged into equivalent process 1' as shown in Figure 6-5. Because the outputs of processes 1, 2, 3 in this example are additive, the output of equivalent 1' is calculated by adding the three outputs of processes 1, 2, 3, and is given in Table 6-7.

Table 6- 7 The output and cost of equivalent process 1'

P_{1j}	30.43	44.21	60.60	75.27	87.13	100.87	115.71
C_{1j}	232	283	334	385	436	487	538
P_{2j}	25.87	37.58	51.51	63.98	74.06	85.74	98.35
C_{2j}	283	334	385	487	538	589	640
P_{3j}	91.29	132.63	181.8	225.81	261.39	302.61	347.13
C_{3j}	1013	1094	1337	1661	1904	/	/
ΣP_{ij}	147.6	214.4	293.9	365.06	422.58	489.2	561.2
ΣFij	1528	1711	2056	2533	2874	/	/

The processes 4 and 5 are two parallel processes and can be merged into equivalent 2' (see Figure 5). The output of 2' is calculated by adding the outputs of processes 4 and 5, and is shown in Table 6-8. Because clay has its own compacting equipment, sand and rock share the same sets of compactors, the ratio of the output of 4 and 5 should be 1:3.85

Table 6- 8 The output and cost of equivalent process 2'

P4j	30.84	61.23	91.78	118.89	148.51
C4j	324	648	972	1296	1620
P5j	118.73	235.74	353.35	457.73	571.76
C5j	1296	1944	3240	4212	/
ΣPij	149.58	296.97	445.13	576.62	720.27
ΣCij	1620	2592	4212	5508	/

Finally, 1' and 2' are two simple series processes and can be merged into one equivalent process. According to the nature of this system, the outputs of the two equivalent process should be equal. Table 6-9 can be calculated from formula (11).

Table 6- 9 Final results of the system

State	1	2	3	4	5
$P_{4'j}$	147.6	214.4	293.9	365.06	422.58
ΣC_{ij}	3148	4303	4648	6745	7086
$\Sigma C_{ij}/\Sigma P_{ij}$	21.33	20.07	15.81	18.48	16.77

The overall optimum state is 3 since it is associated with the least unit production cost.

$$\min_j \{\sum_{i=1',4'} C_{ij} / P_{4'j}\} = 15.81$$

i.e., the optimum production rate of 1' and 2' should be 293.9 m^3/h.

Working backwards, the optimum production rates of processes 1, 2, 3, 4, 5 are 60.60, 51.51, 181.8, 61.23, and 235.74 m^3/h respectively. Processes 1, 2, 3 should be allocated with 4, 5, 11 trucks; processes 4 and 5 should be allocated with 2 and 6 compactors. The total production rate of the system is 293.9 m^3/h, the total operating cost is $4648/h.

In this example, in order to obtain the optimum equipment allocation using the conventional approach, 7*9*14*6*11=58212 simulation runs are required for the entire system; however, by using the method proposed in this paper, only 7+9+11+6+11=47 independent runs for separate processes is required. Moreover, it is very convenient to get simulation results for each process by using sensitivity analysis functions of MicroCYCLONE (Halpin 1990).

In order to verify the results obtained by this method, SLAM II (Pritsker 1986) was used to simulate the entire system under a number of combinations. The model is provided in Figure 6-8 for the interested reader. The identified optimum solutions for three hauling processes are 4, 5, and 11 (or 12) trucks; two compacting processes should be allocated with 2 and 6 compactors separately. The conclusion is the same as provided above.

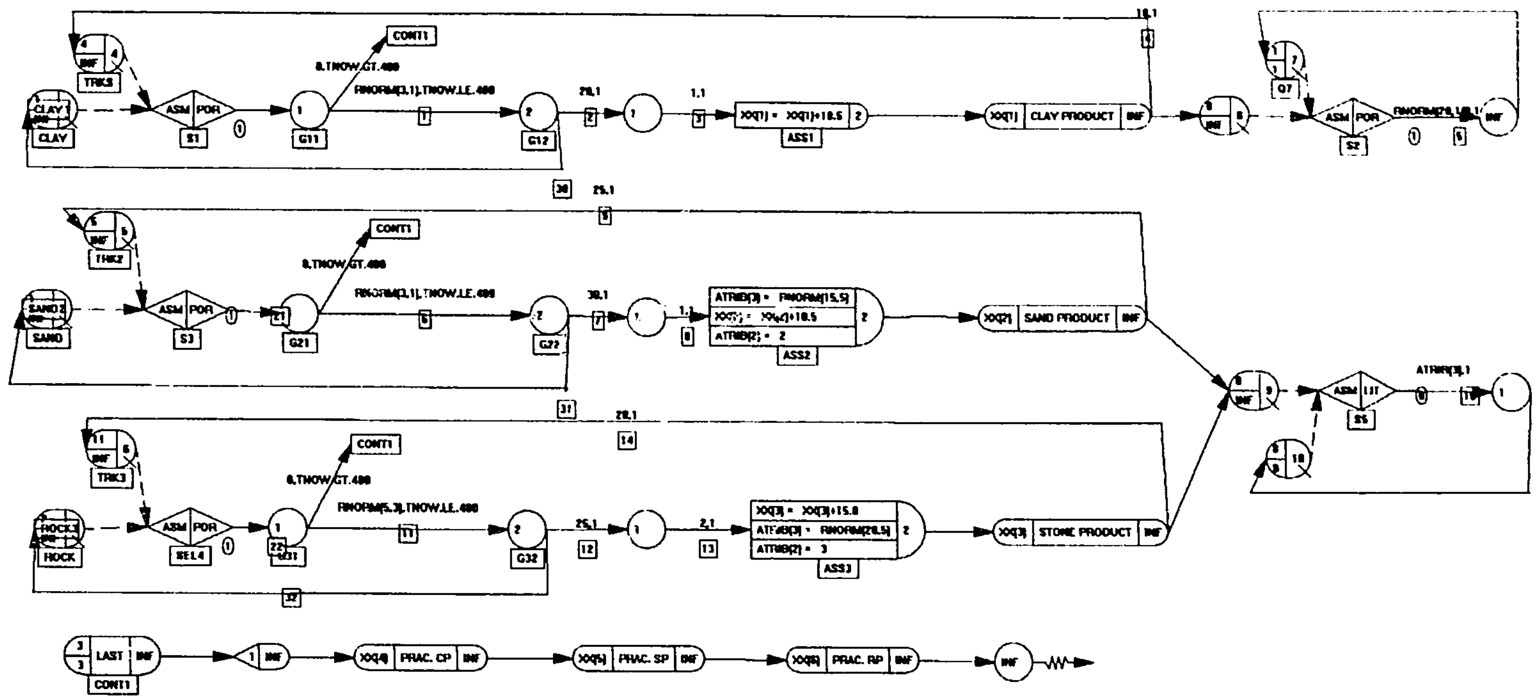


Figure 6- 8 SLAM II model of the earthmoving application

6.5 CONCLUSIONS

The hybrid method presented in his Chapter can locate the optimum solution of a large and complex system by combining simulation and analytic techniques. The benefits from this integration can be summarized into three aspects: 1) making it possible to optimize a large and complex system by using small simulation languages; 2) directly locating the optimum solution; 3) significantly saving simulation runs and computer time.

Since steady simulation results are used for the analysis, the optimum solution obtained from this method may not be the real optimum of a system, although it would be very close. If the selected simulation language permits, the user can take this solution as the initial configuration to do the fine tuning simulation for the entire system by changing decision variables around this initial configuration. These will lead the user to the real optimum solution much faster.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 CONCLUSIONS

This thesis presented the automated process for simulating construction projects in order to simplify simulation and to make it a simple tool for construction practitioners. To substantiate an automated simulation system, this thesis has specifically studied the automated modeling and optimization techniques, which have demonstrated the potential benefits of an automated simulation system. An automated approach is a cost-effective way to assist a project manager in using computer simulation for analyzing and planning construction projects.

The RBM was developed in Chapter 3 as an automated modeling tool mainly for resource-intensive construction projects. It is a general framework which can be specifically implemented to any type of construction projects. A prototype environment with friendly user interface was implemented in Chapter 4 for modeling earthmoving operations by using the RBM methodology. It enables the modeler to build an accurate simulation model for an earthmoving project using resources as the basic building blocks. The modeling process is automated with the user specifying resources and site conditions for a given project. The user does not have to be proficient in simulation as required by the current simulation process. The time required to construct a simulation model for a project by using this RBM environment is very little (going through the resource and project specifications).

The heuristic simulation optimization method presented in Chapter 5 can automatically locate an acceptable resource allocation by optimizing the user specified objectives based on resource utilization. This method is generic and can be implemented to any simulation package to automate the simulation optimization process. As an extension to this automated optimization method, a hybrid method has been studied in Chapter 6 by combining computer simulation with the analytical technique. This approach can use the advantages and avoid the disadvantages of both simulation and mathematical modeling techniques to assist in locating an acceptable solution for a project.

The automated modeling and optimization techniques developed in the thesis have solved two key issues toward a fully automated simulation system.

7.2 RECOMMENDATIONS FOR FURTHER STUDY

This research used SLAM II as the simulation engine. However, SLAM II is a commercial package. We are not allowed access to its code to integrate it with our concepts into a complete automated simulation package. An alternative simulation engine is necessary to replace SLAM II as the embedded simulation engine in order to transfer this research to practical uses by construction practitioners.

The automated optimization techniques presented in Chapter 5 can not guarantee the global optimum. Some more studies such as considering impacts of multiple steps are necessary. More objectives governing automated optimization can be studied, such as maximizing the utilization of key resources.

Mainly system's operating costs are considered at the current stage. This research can be extended from the process level to the project level by considering cost functions (e.g. mobilization and changed cost rates) and other project information (project level constraints).

This research can be further enhanced to integrate it with other computer systems as proposed in Figure 7-1. An intelligent integration platform can be developed to incorporate miscellaneous information for a project including: project design (AutoCAT), site conditions (GIS), project information (WBS), and resources and construction methods. The platform will provide an object-oriented representation for the simulation model of a construction project. An embedded simulation engine is then called to experiment with the generated model. After the automated optimization process, an intelligent reporting module can interpret and report the system's performance. Some of these problems have been dealt with in this thesis research, such as site information, project information and resources. Depending upon the project type, some components need to be further developed.

Design information: Currently many projects are designed using CAD systems (e.g. AutoCAD). The design information should be directly inputted to the integration platform as the important factors to be considered in creating the simulation model for a project.

Site conditions: GIS (Geographical Information System) is a very powerful tool for mapping a construction site, especially for pipeline and mining construction. It accommodates physical components (e.g. road) and associated attributes (e.g. length of a road, surface condition, grade, etc.) to detail a construction site. This information should be directly inputted to the integration platform as the boundary and environmental information of a project.

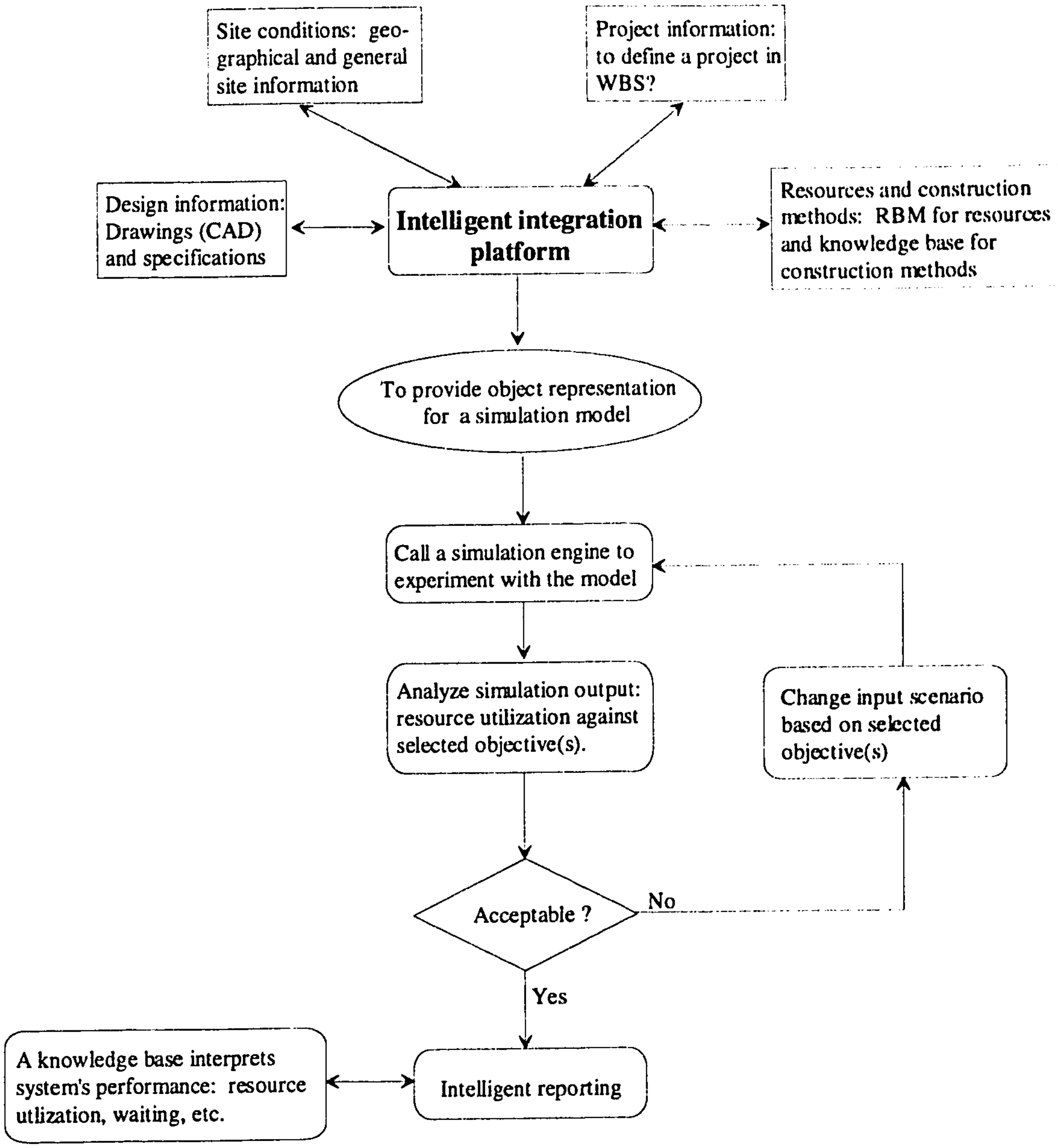


Figure 7-1 An intelligent simulation-based tool for planning construction projects

Project information: HSM (Sawhney and AbouRizk 1995) has been developed to define the formulation of a project in operations and processes by following WBS (work breakdown structure) convention. This approach can be adopted and enhanced to represent a project.

Resources and construction methods: The construction methods used for a project are usually associated with selected resources. For instance, if backhoes are selected for a pipeline project, the trench will have to be dug. RBM has been studied in this thesis. It can be further enhanced to incorporate with a knowledge base to fulfill the requirements of large and complicated construction projects. A library consisted of resources and construction methods should be implemented for each type of construction project.

Integration platform: An intelligent platform is necessary to accomplish the integration of related information. An object-oriented rule-based expert system shell (e.g. M4) can be used to implement this integration module.

Simulation and optimization: A simulation engine can be adopted or developed to accomplish experimentation. The two optimization techniques developed in this thesis can be implemented.

Intelligent reporting: This module should include four major components: 1) to organize simulation results into multiple levels in a representative way of charts or tables to satisfy the requirements of different levels of management (e.g. company level, project level, process level), 2) a knowledge based module to explain complicated statistical simulation results which normally cannot be directly understood by management, 3) a knowledge based module which can evaluate the overall performance of a system by analyzing the resource utilization, the system's production, bottleneck, operating cost and unit production cost based on the work of AbouRizk and Shi (1994), 4) a reporting module which can summarize all above analyses and produce reports required for decision making.

APPENDIX A

BIBLIOGRAPHY

AbouRizk, S.M., Gonzalez-Quevedo A., and D.W Halpin. (1990). "Application of Variance Reduction Techniques in Construction Simulation", MicroComputers in Civil Engineering, ELSEIER Science Publishers Vol.5, 299-306.

AbouRizk, S. M. and D. W. Halpin (1991). "Visual Interactive Fitting of Beta Distributions." *J. Constr. Engrg. & Mgmt.*, ASCE, 117(4), 589-605.

AbouRizk, S. M. and D. W. Halpin (1992). "Statistical Properties of Construction Duration Data." *J. Constr. Engrg. & Mgmt.*, ASCE, 118(3), 525-544.

AbouRizk, S. M. (1994). Technical Discussion on: "Modeling Operational Activities in Object-Oriented Simulation." by A. A. Oloufa. *J. of Computing in Civil Engineering*, ASCE.

AbouRizk, S.M. and J. Shi (1994). "Automated Construction Simulation Optimization." *J. Constr. Engrg. & Mgmt.*, ASCE, 120(2), 374-385.

Ahuja, H. N. and Nandakumar (1985). " Simulation Model to Forecast Project Completion Time." *J. Constr. Engrg. & Mgmt.*, ASCE, 111(4), 325-342).

Alkass, S. and F. Harris (1988). "Expert System for Earthmoving Equipment Selection in Road Construction." *J. Constr. Engrg. & Mgmt.*, ASCE, Vol. 114, No.3, 426-440.

Amirkhanian, S. and N. J. Baker (1992). "Expert System for Equipment Selection for Earth-moving operations." *J. Constr. Engrg. & Mgmt.*, ASCE, Vol. 118, No.2, 318-331.

Andradottir, Sigrun (1992). "Discrete Optimization in Simulation: A Method and Application." *Proceedings of the 1992 Winter Simulation Conference*, 483-486.

Arditi, D. (1983). "Diffusion of Network Planning in Construction." *J. Constr. Div.*, ASCE, 109(1), 1-12.

Arditi, D. and M. Z. Albulak (1986). "Line-of-Balance Scheduling in Pavement Construction." *J. Constr. Engrg. & Mgmt.*, ASCE, 112(3), 411-424.

Arditi, D. (1985). "Construction Productivity Improvement." *J. Constr. Engrg. & Mgmt.*, ASCE, 111(1), 1-14.

Azadivar, F. (1992). "A Tutorial On Simulation Optimization." *Proceedings of 1992 Winter Simulation Conference*, 198-204.

Balci, O. and R. E. Nance (1992). "The Simulation Model Development Environment: An Overview." *Proceedings of the 1992 Winter Simulation Conference*, 762-736.

Bernold, L. and D.W. Halpin (1984). "Microcomputer Cost Optimization of Earth Moving Operations." Vol.2, Proceedings, 4th International Symposium on Organization of Construction, University of Waterloo, Waterloo, Ontario, July.

Bernold, L. (1989). "Simulation of Nonsteady Construction Processes." *J. Constr. Engrg. & Manag.*, ASCE, June, Vol.115(2), 163-178.

Berrios, J. and D.W. Halpin (1988). "Construction Optimization Using Resource Simulation." Concrete International - Design and Construction, June, Vol.10, No.6, pp.38-42.

Biles, W.E. (1974). "A Gradient Regression Search Procedure for Simulation Experiment." *Proceedings of the 1974 Winter Simulation Conference*, 491-497.

Bortscheller, B. J. and E. T. Saulnier (1992). "Model Reusability in A Graphical Simulation Package." *Proceedings of the 1992 Winter Simulation Conference*, 764-772.

Business Roundtable, (1982). "*Measuring Productivity in Construction. Construction Industry Cost Effectiveness Report.*" Business Roundtable. New York.

Carr, R.I., and W. L. Meyer (1974). "Planning Construction of Repetitive Building Units." *J. Constr. Div.*, ASCE, 100(3), 403-412.

Chang, D (1987), RESQUE, PhD thesis, University of Michigan, Ann Arbor, Mich.

Choromckos, J. and K. E. McKee (1981). "Construction Productivity Improvement." *J. Constr. Div.*, ASCE, 107(1), 35-48.

Christian, J. and Caldera, H. (1988). "Earthmoving Cost Optimization by Operational Research." *Can. J. Civ. Engrg.*, Vol 15, No.4, 679-684.

Chrzanowski, E. N. and D. W. Johnston (1986). "Application of Linear Scheduling." *J. Constr. Engrg. & Mgmt.*, ASCE, 477-491.

Cohenca, D., A. Laufer, and W.B. Ledbetter (1989). "Factors Affecting Construction Planning Efforts." *J. Constr. Engrg. & Mgmt.*, ASCE. 115(1), 70-87.

Cole, L.J.R. (1991). "Construction Scheduling: Principles, Practices, and Six Case Studies." *J. Constr. Engrg. & Mgmt.*, ASCE, 117(4), 579-588.

Crandall, K. C. (1977). "Analysis of Schedule Simulations." *J. Constr. Div.*, ASCE, 103(3), 387-294.

Crandall, K. C. and J. C. Woolery (1982). "Schedule Development Under Stochastic Scheduling." *J. Constr. Div.*, ASCE, 108(2), 321-329.

Dressler, J. (1974). "Stochastic Scheduling of Linear Construction Sites." *J. Constr. Div.*, ASCE, 100(4), 175-195.

Easa, S.M. (1987). "Earthwork Allocations with Nonconstant Unit Costs." *J. Constr. Engrg. & Mgmt.*, ASCE, Vol. 113, No.1, 34-50.

Glynn, P.W. (1986) Optimization of Stochastic Systems, *Proceedings of the 1986 Winter Simulation Conference*, 366-375.

Goldsman, D., B.L.Nelson, and B. Schmeiser (1991). "Methods for Selecting the Best System." *Proceedings of the 1991 Winter Simulation Conference*, 177-186.

Grain, R.C., J.M. Sussman, T.J.Schriber, J.O. Henriksen, and S.D. Roberts (1992). "The Winter Simulation Conference: Celebrating Twenty-Five Years of Progress." *Proceedings of the 1992 Winter Simulation Conference*, Arlington, VA, 3-36.

Halpin, D.W. and Woodhead, R.W (1976). "Design of Construction and Process Operations." John Wiley and Sons, Publishers, Inc., New York, N.Y.

Halpin, D.W. (1977). "CYCLONE - A Method for Modeling Job Site Processes." *J. Constr. Div.*, ASCE, 103(3), 489-499.

Halpin, D.W. (1985). "Resource Analysis Using Microcomputers - Final Report, NSF Grant CEE-8208271." School of Civil Engineering, Georgia Institute of Technology, Atlanta, GA.

Halpin, D.W. AbouRizk and Hijazi, A (1989). "Sensitivity Analysis of Construction Operations." *Proceedings of the 7th National Conference on Microcomputers in Civil Engineering*, Orlando, Florida, 181-185.

Halpin, D. W. (1990). "MicroCYCLONE User's Manual." Division of Construction Engineering and Management, Purdue University, West Lafayette, Indiana.

Handa, V. K. and R. M. Barcia (1986). "Linear Scheduling Using Optimal Control Theory." *J. Constr. Engrg. & Mgmt.*, ASCE, 112(3), 387-393.

Huang, R., A.M Grigoriadis, and D. W. Halpin (1994). "Simulation of Cable-stayed Bridges Using DISCO." *Proceedings of the 1994 Winter Simulation Conference*, 1130-1136.

Ibbs, C.W. (1987). "Future direction for computerized construction research." *J. Constr. Engrg. & Mgmt.*, ASCE. 112(3), 326-345.

Ioannou, P.G (1989). "UM_CYCLONE user's guide." Dept. of Civil Engineering, The University of Michigan, Ann Arbor, Mich.

Jaafari, A. (1984). "Critism of C.P.M. for Project Planning Analysis." *J. Constr. Engrg. & Mgmt.*, ASCE. 110(2), 222-233.

Johnston, D. W. (1981). "Linear Scheduling Method for Highway Construction." *J. Constr. Div.*, ASCE, 107(2), 247-261.

Koehn, E. and S. Caplan (1985). "Work Improvement Data for Small and Medium Size Contractors." *J. Constr. Engrg. & Mgmt.*, ASCE, 113(2), 327-339.

Liu, L. Y. and P. G. Ioannou (1992). "Graphical Object-Oriented Discrete-Event Simulation System." *Proceedings of the 1992 Winder Simulation Conference*, 1285-1291.

Luna, J. (1991). "Object Framework for Application of Multiple Analysis Paradigms." *Object-Oriented Simulation 1991*, Simulation Series Volume 23, Number 3, 81-86.

Luna, J.L. (1992). "Hierarchical, Modular Concepts Applied to an Object-Oriented Simulation Model Development Environment." *Proceedings of the 1992 Winter Simulation Conference*, 694-699.

Martinez, J. and P. G. Ioannou (1994). "General Purpose Simulation with Stroboscope." *Proceedings of the 1994 Winter Simulation Conference*, 1159-1166.

McCahill, D.F. and L. E. Bernold (1993). "Resource-Oriented Modeling and Simulation in Construction." *J. Constr. Engrg. & Mgmt.*, ASCE, 119(3), 590-606.

McCough, E.H. (1982). "Scheduling: Effective Methods and Techniques." *J. Constr. Div.*, ASCE. 108(1), 75-84.

Meketon, M.S. (1987). "Optimization in Simulation: A Survey of Recent Results." *Proceedings of the 1987 Winter Simulation Conference*, 58-67.

Mohieldin, Y.A. (1989). "Analysis of Construction Processes." Ph.D. Dissertation, Civil Engineering Dept., University of Maryland, College Park, MD, May.

Moselhi, O. and K. Ei-Rayes (1993). "Scheduling of Repetitive Projects with Cost Optimization." *J. Constr. Engrg. & Mgmt.*, ASCE, 119(4), 681-697.

Nunnally, S.W. (1980). *Construction Methods and Management.* Prentice_Hall, Inc., Englewood Cliffs, New Jersey.

Oloufa, A.A. (1993). "Modeling Operational Activities in Object-Oriented Simulation." *Journal of Computing in Civil Engineering*, ASCE, Vol. 7, No. 1, 94-106.

Paul, R. J. (1992). "The Computer Aided Simulation Modeling Environment: An Overview." *Proceedings of the 1992 Winter Simulation Conference*, 737-746.

Paulson, B. C. (1973). "Project Planning and Scheduling: Unified Approach." *J. Constr. Div.*, ASCE, 99(1), 45-58.

Paulson, B.C. Jr., (1978). "Interactive Graphics for Simulating Construction Operations." *J. Constr. Div.*, ASCE, 104(1), 69-76.

Paulson, B.C., Jr., Chan, W.T., Koo, C.C. (1987). "Construction Operations Simulation by Microcomputer." *J. Constr. Engrg. & Mgmt.*, ASCE, 113(2), 302-314.

Pegden, C.D., and M.P. Gately (1977). "Decision Optimization for GASP IV Simulation Models." *Proceedings of the 1977 Winter Simulation Conference*, 125-133.

Perera, S. (1982). "Resource Sharing in Linear Construction." *J. Constr. Div.*, ASCE, 109(1), 102-111.

Peurifoy, R.L., and W.B. Ledbetter (1985). *Construction Planning, equipment and methods.* McGraw-Hill, New York, N.Y.

Pritsker, A.A.B. (1986). "*Introduction to Simulation and SLAM II: Networks for decision Support.*" Prentice Hall, Englewoods Cliffs, New Jersey.

Pultar, Mustafa (1990). "Progress-Based Construction Scheduling." *J. Constr. Engrg. & Mgmt.*, ASCE, 116(4), 671-688.

Pritsker, A. A. B. (1986), "Introduction to Simulation and SLAM-II", John and Sons, Inc., New York, N.Y.

Riggs, L. (1979), "Sensitivity Analysis of Construction Operations", Ph.D. Thesis, School of Civil Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.

Robbins, H. and S. Monro (1951). "A Stochastic Approximation Method." *Annals of Mathematical Statistics*, 22:400-407.

Russell, A.D. (1985). "Microcomputers, Management anf Highrise Construction: the Next Step." *Can. J. Civ. Engrg.*, 12(2), 396-414.

Russel, A. D. and W. F. Caselton (1988). "Extension to Linear Scheduling Optimization." *J. Constr. Engrg. & Mgmt.*, ASCE, 114(1), 36-52.

Sarraj Z. M. A. (1990). "Formal Development of Line-of-Balance Technique." *J. Constr. Engrg. & Mgmt.*, ASCE, 116(4), 689-704.

Sawhney, A. And S.M. AbouRizk (1995). "Simulation based planning method for construction project." Accepted for publication in the *J. Constr. Engrg. & Mgmt.*, ASCE.

Selinger, S. (1980). "Construction Planning for Linear Projects." *J. Constr. Div.*, ASCE, 106(2), 195-205.

Shi, J., G. Yuang, and J. Zuo (1987). "Computer Simulation Study on the Selection of Construction equipment for High Embankment Dams." *J. of Hydroelectric Engrg.*, No.3, 62-72.

Shi, J. And S. AbouRizk (1994). "A Resource_based Simulation Approach with Application in Earthmoving/Strip Mining." *Proceedings of the 1994 Winter Simulation Conference*, Florida, U.S.A., 1124-1129.

Shi, J. and R. Wales (1994). " *An Introduction to Simulation with a Case Study from the Syncrude Reclamation Project.*" Submitted to North American Construction Group. Department of Civil Engineering, University of Alberta.

Shi, J. and S. M. AbouRizk (1995). "An Optimization Method for Simulating Large and Complex Systems." *Engineering Optimization* accepted for publication.

Sidwell, A.C., and L.J.R. Cole (1989). "*Computers in Construction Managemant, the final report.*" South Australian Inst. Of Tech., Adelaide, Australia.

Smith, D.E. (1976). "Automatic Optimum-seeking Program for Digital Simulation." *Simulation*, 27:27-32.

Srilal Perera (1982). "Resource Sharing in Linear Construction." *J. Constr. Engrg. & Mgmt.*, ASCE, March, Vol. 109(1), 102-111.

Stradal, O. and J. Cacha (1982). "Time Space Scheduling Method." *J. Constr. Div.*, ASCE, 108(3), 445-457.

Tavakoli, A. (1985), "Productivity Analysis of Construction Operations", *J. Constr. Engrg. & Mgmt.*, Vol.111, No.1, March, 31-39.

Tommelein, I. D., and A. M. Odeh (1994). "Knowledge-based Assembly of Simulation Networks Using Construction Designs, Plans, and Methods." *Proceedings of the 1994 Winter Simulation Conference*, 1145-1158.

Touran, A. (1987). "Simulation of Tunnel Operation", *J. Constr. Engrg. & Mgmt.*, ASCE, 113(4), 554-568.

Touran, A. (1987) "Simulation of Tunnel Operation." *J. Constr. Engrg. & Mgmt., ASCE*, 113(4), 554-568.

Vanegas, J.A., E.B. Bravo, and D.W. Halpin (1993). "Simulation Technologies for Planning Heavy Construction Processes." *J. Constr. Engrg. & Mgmt.*, ASCE, Vol.119(2), 336-354.

Pritsker, A. A. B. (1986), "Introduction to Simulation and SLAM-II", John and Sons, Inc., New York, N.Y.

Yan, D., and H. Mukai (1992). "Stochastic Discrete Optimization." *SIAM Journal on Control and Optimization* 30:594-612.

Zeigler, B.P. (1976). "Theory of Modelling and Simulation." Wiley, N.Y. (Reissued by Krieger Pub. Co., Malabar, Fla. 1985).

Zeigler, B.P. (1984). "Multifaceted Modelling and Discrete Event Simulation." Academic Press, London and Orlando, Fla.

Zeigler, B.P. (1985). "System-theoretic Representation of Simulation Models." *IIE Transactions*, 16, no.1:19-34.

Zeigler, B.P. (1987). "Hierarchical, modular discrete-event modelling in an object-oriented environment." *Simulation*, 49(5), 219-230.

APPENDIX B

SLAM II STATEMENTS OF ATOMIC MODELS

```
EA1   AWAIT(QN1),EXCAVATOR/1,1;
      ACTIVITY/EX1, ET1, ,EG1;Fxcavate;
EG1   GOON,1;
      ACTIVITY/EX2, ET2, ,EG1;swing;
EG2   GOON,1;
      ACTIVITY()/EX3, ET3, ,EG2;dump;
EG3   GOON, 2;
      ACTIVITY, , , CNT;
LNK   ACTIVITY,,,LI1;
CNT   ASSIGN, SS(I)=SS(I)+MES(I),1;
      ACTIVITY, , SS(I).GT.OBJ, FIN;
      ACTIVITY/LD4, LT4,SS(I).LT.OBJ,EF1;return;
EF1   FREE,EXCAVATOR/1,1;
      ACTIVITY, , , EA1;
      RESOURCE, EXCAVATOR(CAP),QN1;
```

Figure B-1 SLAM II network statement of excavation *r-process*

```
LI1   AWAIT(QN1),LOADER/1,1;
      ACTIVITY/LD1, LT1, ,LG1;Excavate;
LG1   GOON,1;
      ACTIVITY/LD2, LT2, ,LQ2;swing;
LQ2   QUE(QN2), 0, , ,LS1;
LI2   UNBATCH,1, ;
      ACTIVITY, , ,LQ3;
LQ3   QUE(QN3),INI, , ,LS1;
LS1   SELECT, ASM, POR, ,LQ2,LQ3;
      ACTIVITY()/LD3, LT3, ,LG2;dump;
LG2   GOON, 2;
      ACTIVITY, , , CNT;
      ACTIVITY,,,LA1;
LA1   ACCUMULATE,ATRIB(1),ATRIB(1),LAST, ;
DNK   ACTIVITY,,,HI1;
CNT   ASSIGN, SS(I)=SS(I)+MES,1;
      ACTIVITY, , SS(I).GT.OBJ, FIN;
      ACTIVITY/LD4, LT4,SS(I).LT.OBJ,LQ1;return;
LQ1   FREE,LOADER/1,1;
      ACTIVITY, , , SINK;
LD1   RESOURCE, LOADER(CAP),QN1;
```

Figure B-2 SLAM II network of loading *r-process*

```
SA1  AWAIT(QN1),TRACTOR/1,1;
     ACTIVITY/SC1,ST1, ,SG1;LOADING;
SG1  GOON,2;
     ACTIVITY/SC2, ST2, , SG2;MOVE TO DUMP;
     ACTIVITY/SC5,ST5, ,SF1;TRACTOR REPOSITION;
SF1  FREE,TRACTOR/1,1;FREE TRACTOR;
     ACTIVITY, , ,SINK;
SG2  GOON,1;
     ACTIVITY/SC3,ST3, , SG3;
SG3  GOON,2;
DNK  ACTIVITY, , , SI1;
     ACTIVITY, , , CNT;
CNT  ASSIGN, SS(I)=SS(I)+MES, 1;
     ACTIVITY, , SS(I).GE.OBJ, FIN;
     ACTIVITY/SC4, ST4,SS(I).LT.OBJ, SA1;RETURN TO NEXT CYCLE;
RES  RESOURCE,TRACTOR(CAP1),QN1;
```

Figure B-3 SLAM II network statement of scraping *r-process*

```
HI1  GOON,1;
     ACTIVITY/HL1, HT1, , HQ1;
HQ1  AWAIT(QN1),DUMP/1,1;
     ACTIVITY/HL2,HT2, , HG1;
HG1  FREE, DUMP/1, 2;
LNK  ACTIVITY, , , VI1;
     ACTIVITY, , , CNT;
CNT  ASSIGN, SS(I)=SS(I)+MES, 1;
     ACTIVITY, , SS(I).GE.OBJ, FIN;
     ACTIVITY/HL3, HT3,SS(I).LT.OBJ, HQ2;
HQ2  QUE(QN2),TRUCK,TRUCK, , ;
     ACTIVITY, , , HA1;
HA1  ASSIGN,ATRIB(1)=MES(I)/MES(J);
DNK  ACTIVITY, , ,LI2;
RES  RESOURCE,DUMP(CAP),QN1;
```

Figure B-4 SLAM II network statement of hauling *r-process*

```
RI1  AWAIT (PN1),TRAC/1,1;
     ACTIVITY/SP1,RT1,,RG1;positioning;
RG1  GOON,1;
     ACTIVITY/SP2,RT2,,RG2;cut;
RG2  GOON,1;
     ACTIVITY/SP3,RT3,,RG3;move
RG3  GOON,1;
     ACTIVITY/SP4,RT4,,RG4;dump;
RG4  GOON,2;
     ACTIVITY, , , CNT;
DNK  ACTIVITY,,,CI1;
CNT  ASSIGN, SS(I)=SS(I)+NES,1;
     ACTIVITY, , SS(I).GE.OBJ, FIN;
     ACTIVITY/SP5,RT5,SS(I).LT.OBJ,RF1;return;
RF1  FREE,TRAC,1;
     ACTIVITY, , , SINK;
RES  RESOURCE,TRACT(CAP);
```

Figure B-5 SLAM II network statement of spreading *r-process*

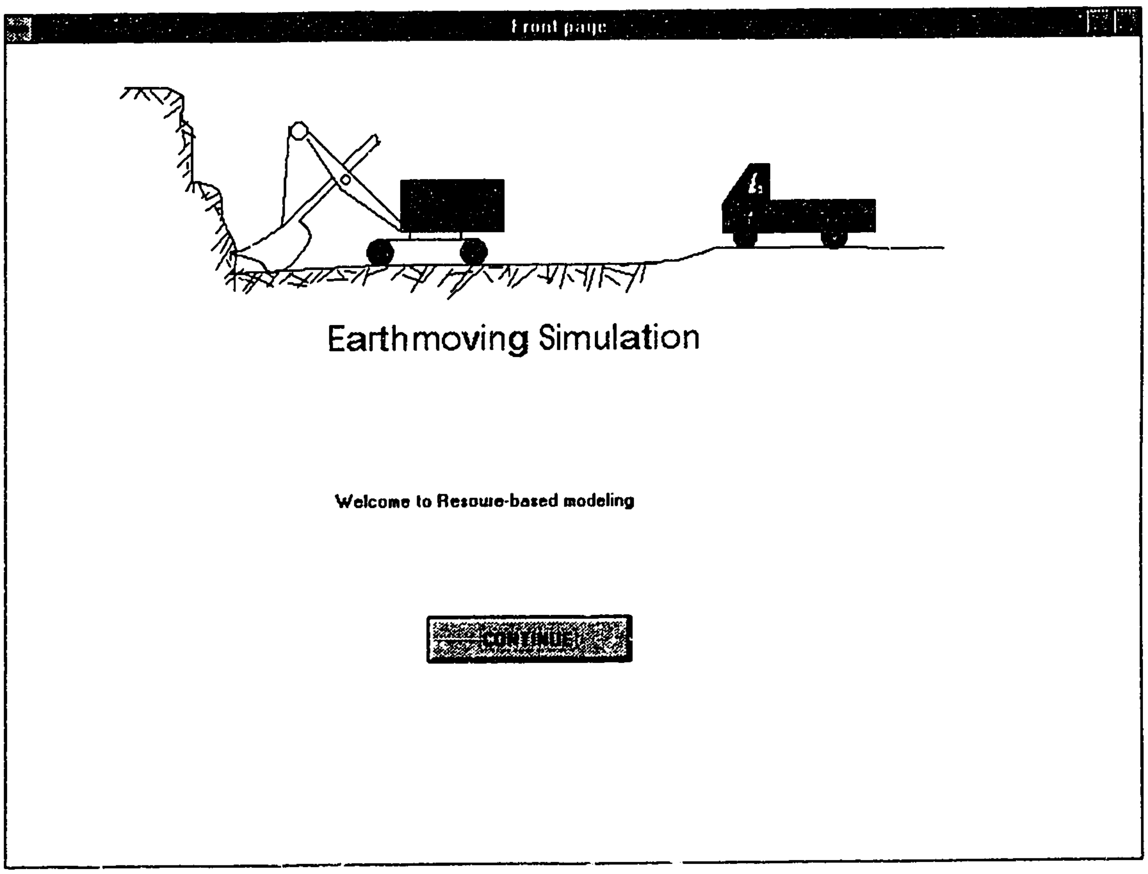
```
CI1  ACCUMULATE,X,X,LAST, ;
     ACTIVITY, , ,CA1;
CA1  AWAIT(CN1),COMPACTOR/1,1;
     ACTIVITY/CP1,CT1, ,CG1;POSITIONING
CG1  GOON, ;
     ACTIVITY/CP2,CT2, ,CF1;COMPACTION
CF1  FREE,COMPACTOR/1,1;
     ACTIVITY, , ,CS1;
CNT  ASSIGN,SS(I)=SS(I)+MES;
     ACTIVITY, , ,CG2;
CG2  GOON,1;
     ACTIVITY, ,SS(I).GE.OBJ, FIN;
     ACTIVITY, ,SS(I).LT.OBJ, SINK;
     RESOURCE,COMPACTOR(CAP),CN1;
```

Figure B-6 SLAM II network statement of compaction *r-process*

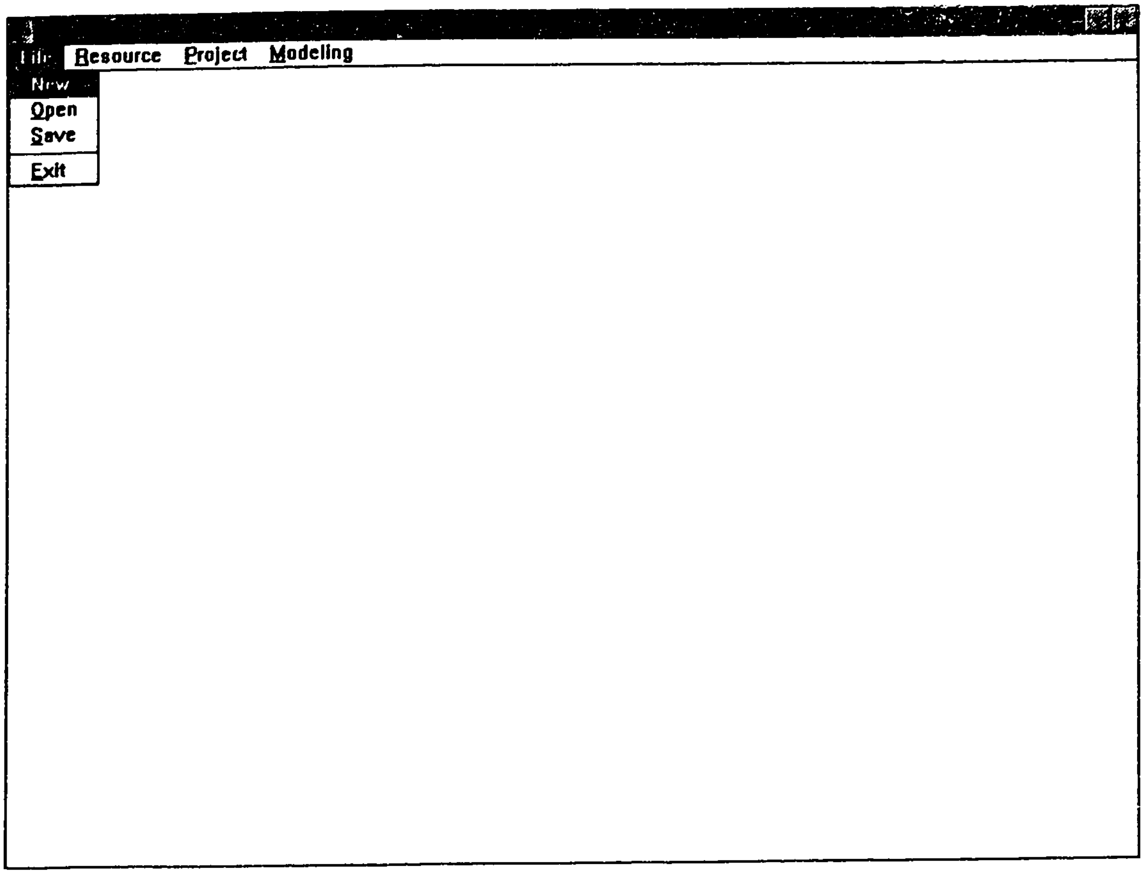
APPENDIX C

SYSTEM MANUAL FOR THE MODELING ENVIRONMENT OF EARTHMOVING OPERATIONS

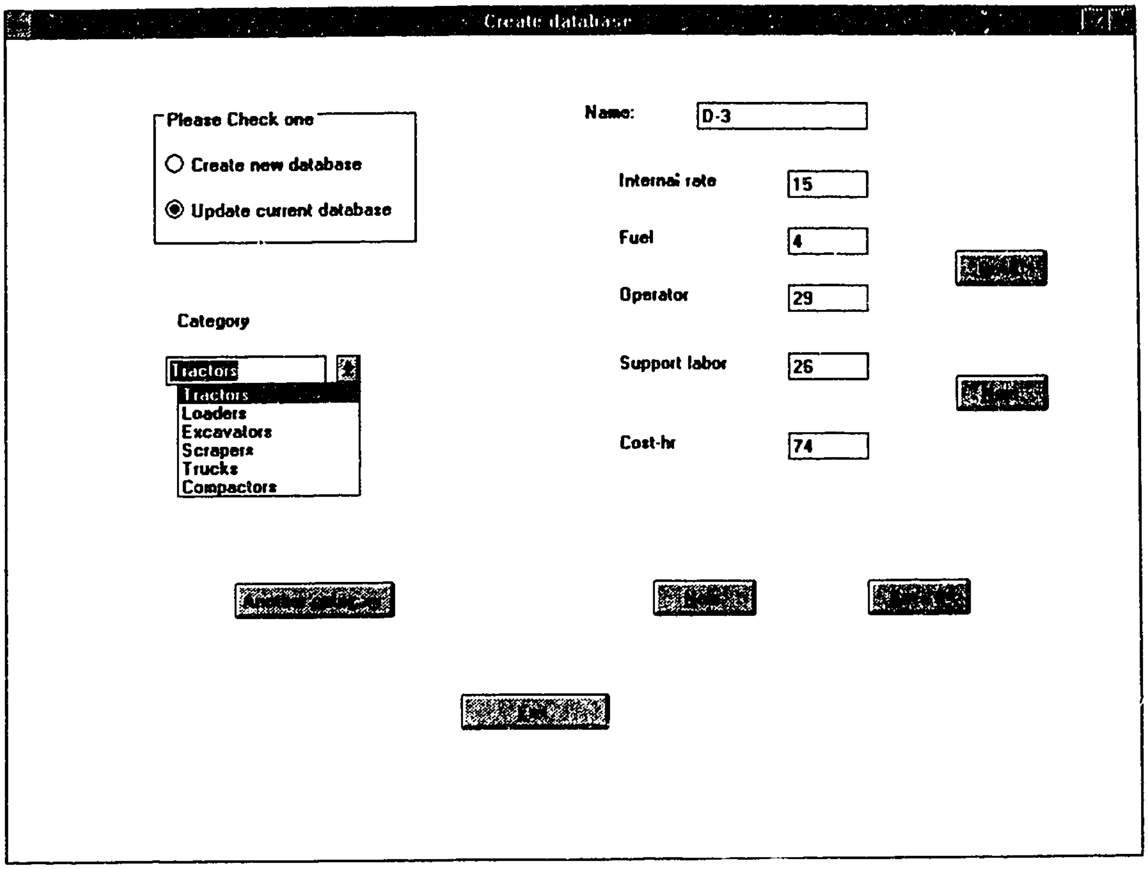
This automated modeling environment was illustratively implemented for earthmoving operations based on the resource-based modeling concepts presented in Chapter 3 and Chapter 4 using VISUO-BASIC programming language. This system manual shows the major user-interface screens and explains them.



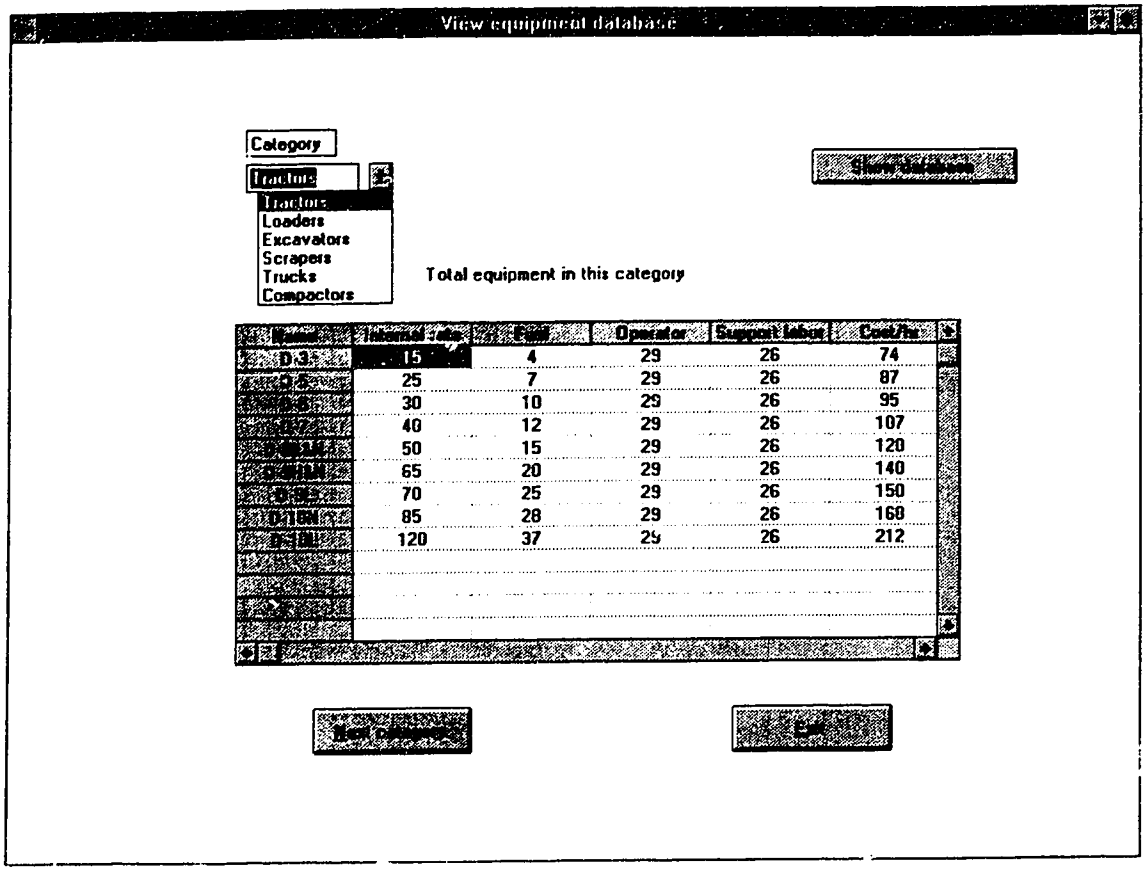
This is the first screen when the user starts running the automated modeling environment for earthmoving operations.



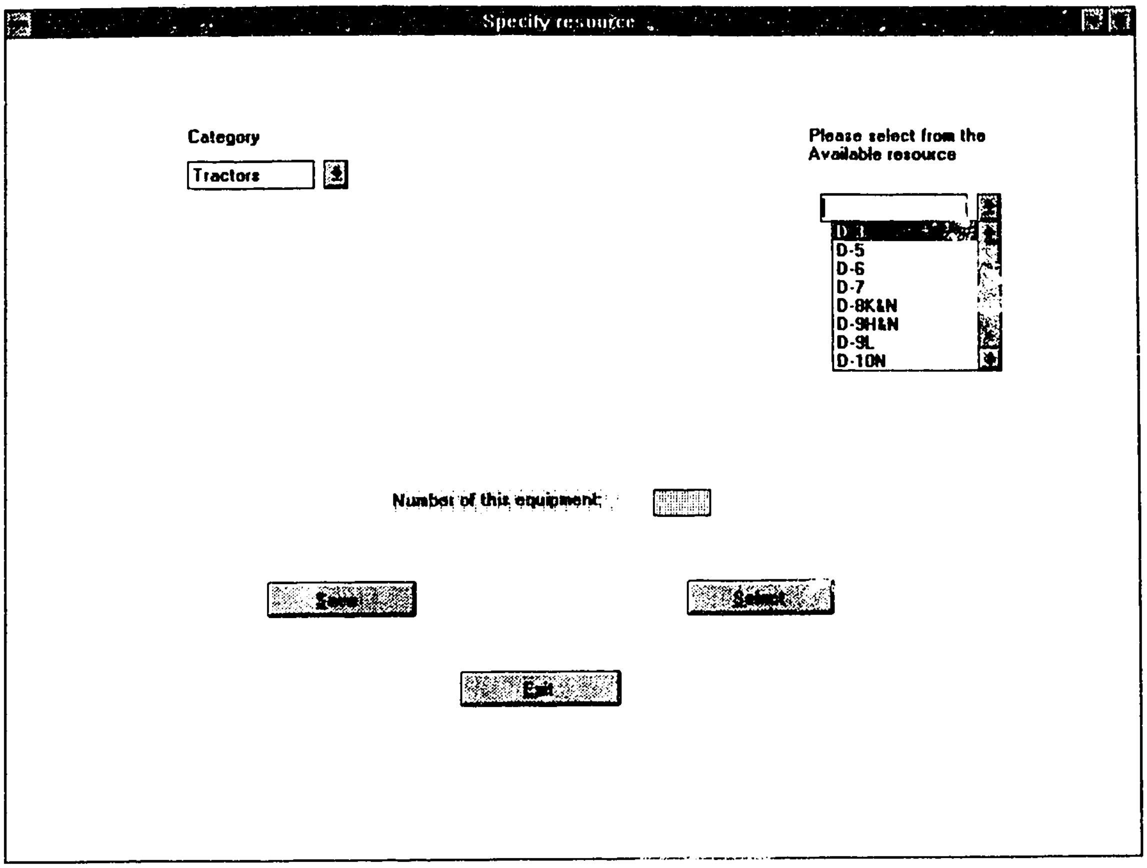
The system will move to this page after the user clicks the "CONTINUE" button as shown on the first screen. This is the main menu screen of the environment. A new project is initialized, an old project can be opened, and a project can be saved through this screen. The user is lead to resource specification, project specification, or modeling functions by clicking proper menu as shown on this screen.



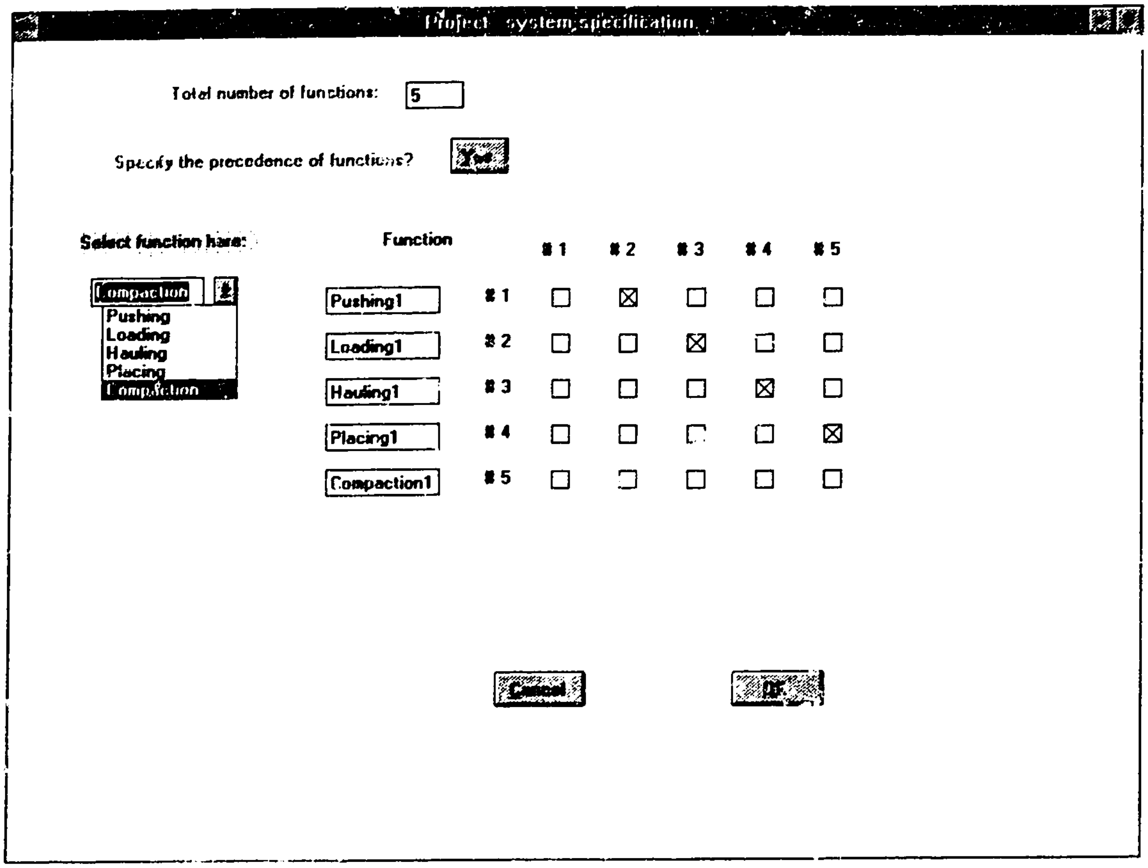
This screen can be activated with the user selecting "Create database" item of the "Resource" menu as shown on the main menu screen. It lets the user update and create the equipment database to reflect a company's situation. The information for each piece of equipment includes its category, name (model), internal rate, fuel cost, operator cost, support labor cost, and total cost.



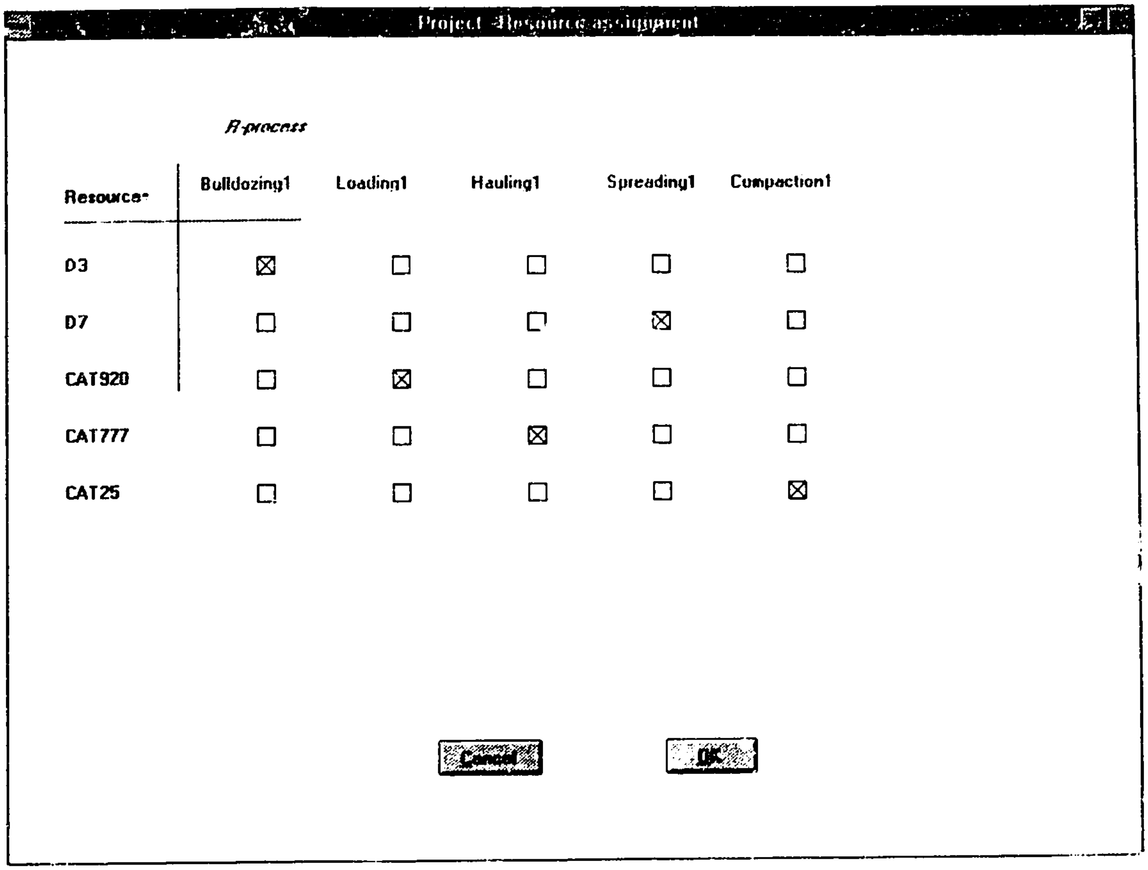
This "View equipment data" screen can be obtained by selecting "View equipment database" item of "Resource" menu as shown on the main menu screen. It allows the user to view the database system previously created.



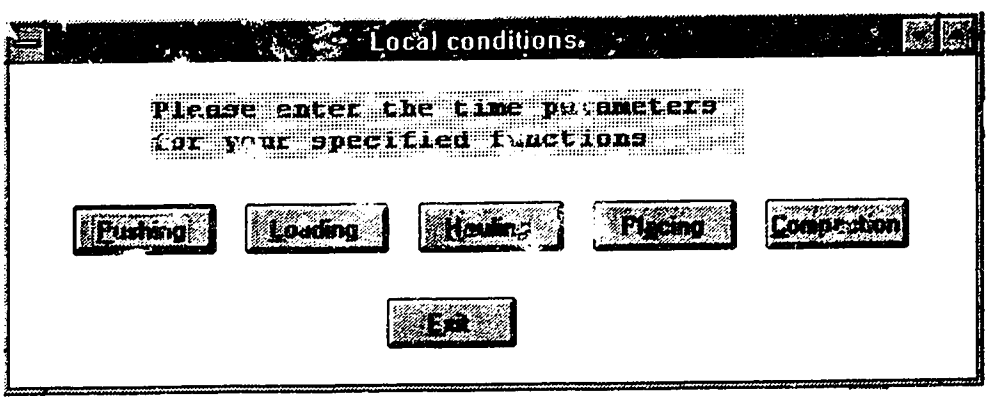
This screen lets the user specify resources required for a specific project. The user should first select the category (e.g. Tractors). Then a specific name (or model) can be specified (e.g. D10N). The number required is also specified by the user in the given box. After the user clicks the "Select" button, the specified resource will be kept in a temporary array. After the last resource has been specified, the user can clicks "Save" button. The resource combination will be permanently saved.



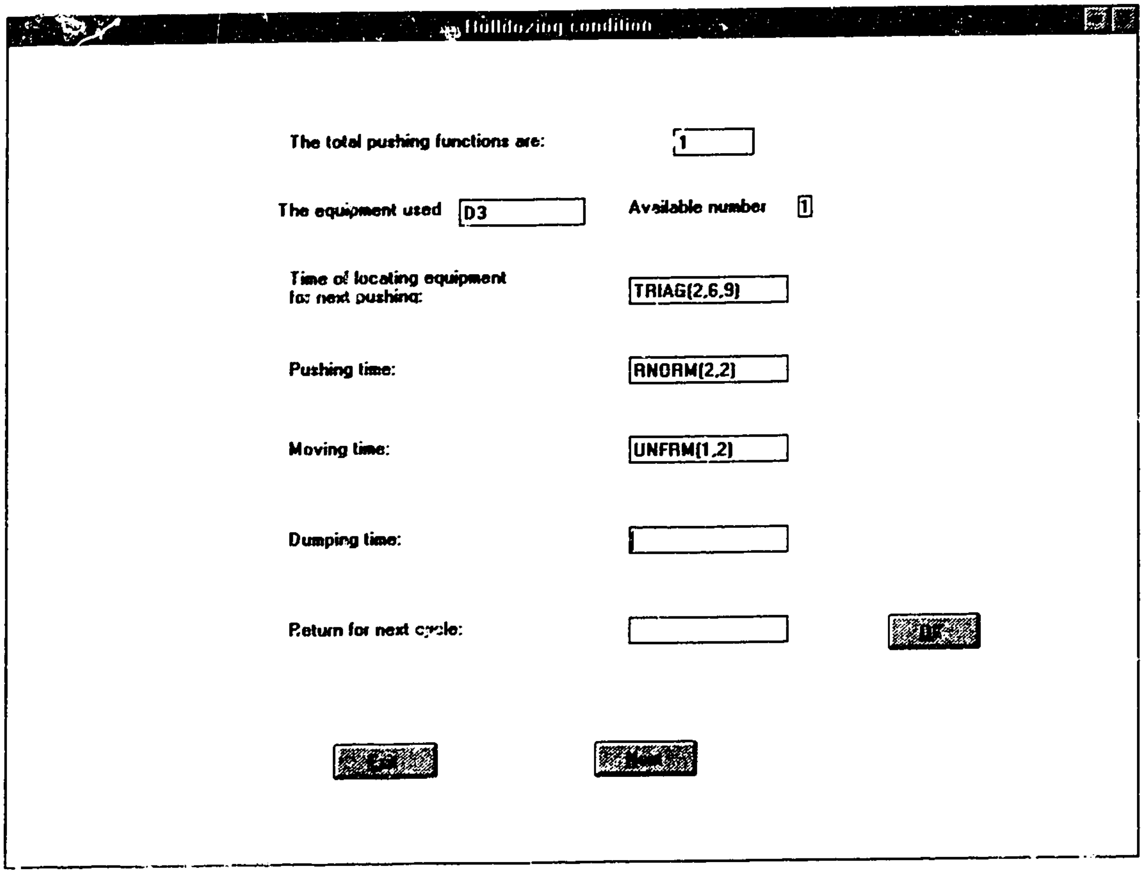
This screen is the first part of the "project" specification as shown on the main menu screen. It allows the user to represent the formulation of a project in *r-process.* The logical sequences among specified *r-processes* is identified by specifying the follows of each *r-process*. Checking an *r-process* along its vertical axis indicates that it follows the *r-processes* referenced by the horizontal axis.



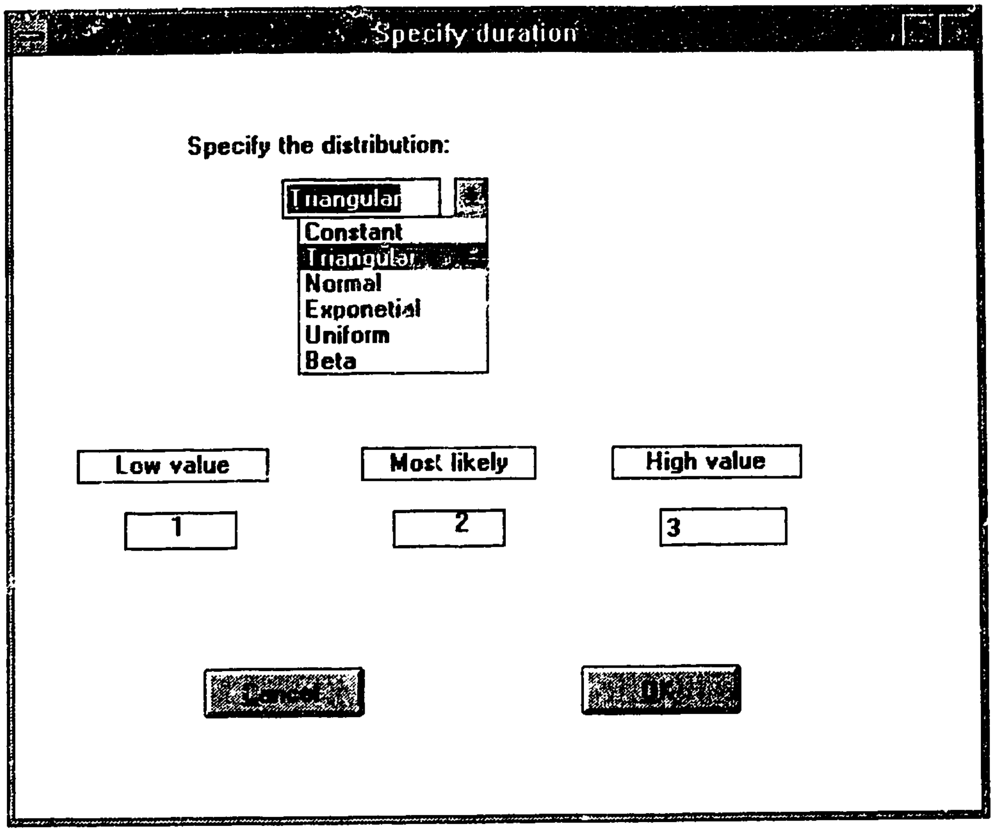
This screen is the second part of the project specification. The user is required to assign resources to *r-processes*. A resource can be assigned to multiple *r-processes* (shared resource). Multiple resources can also be assigned to one single *r-process*.



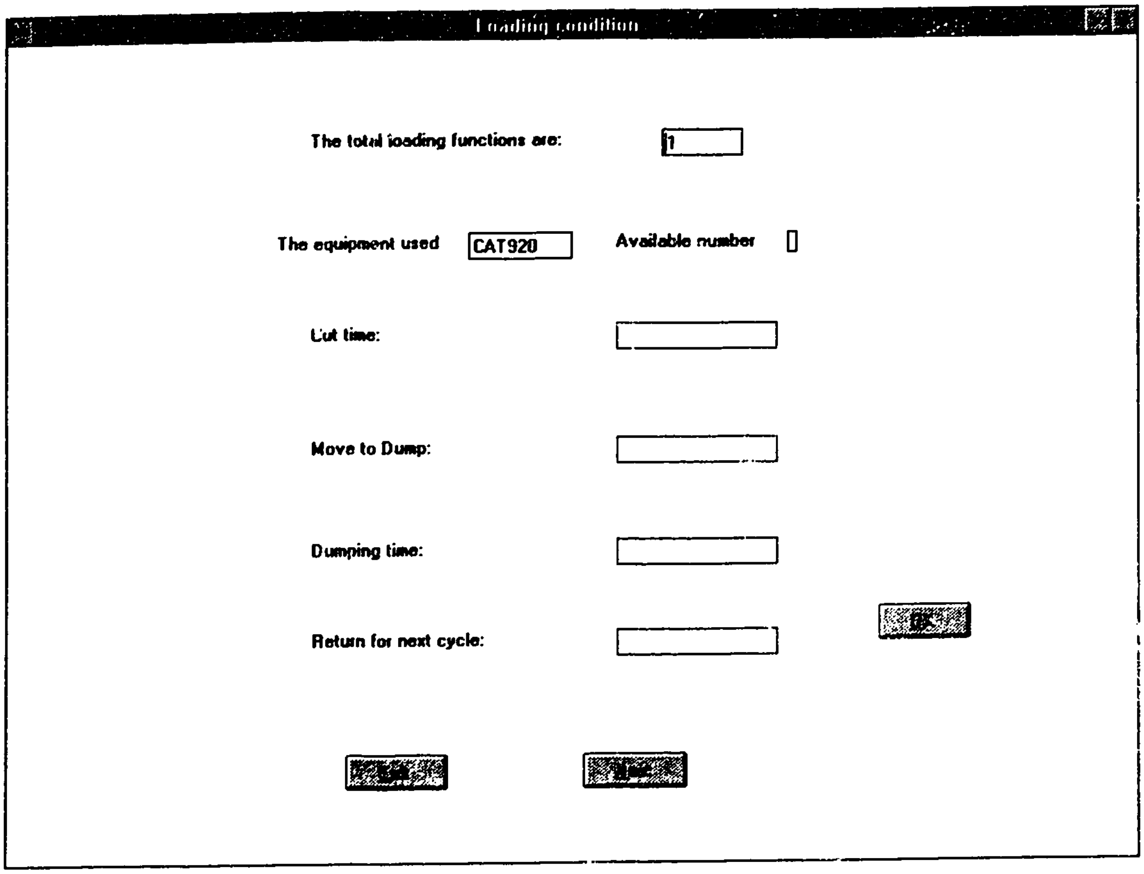
Local conditions are the third part of the project specification. The user can input activity durations for each *r-process* activated from this screen.



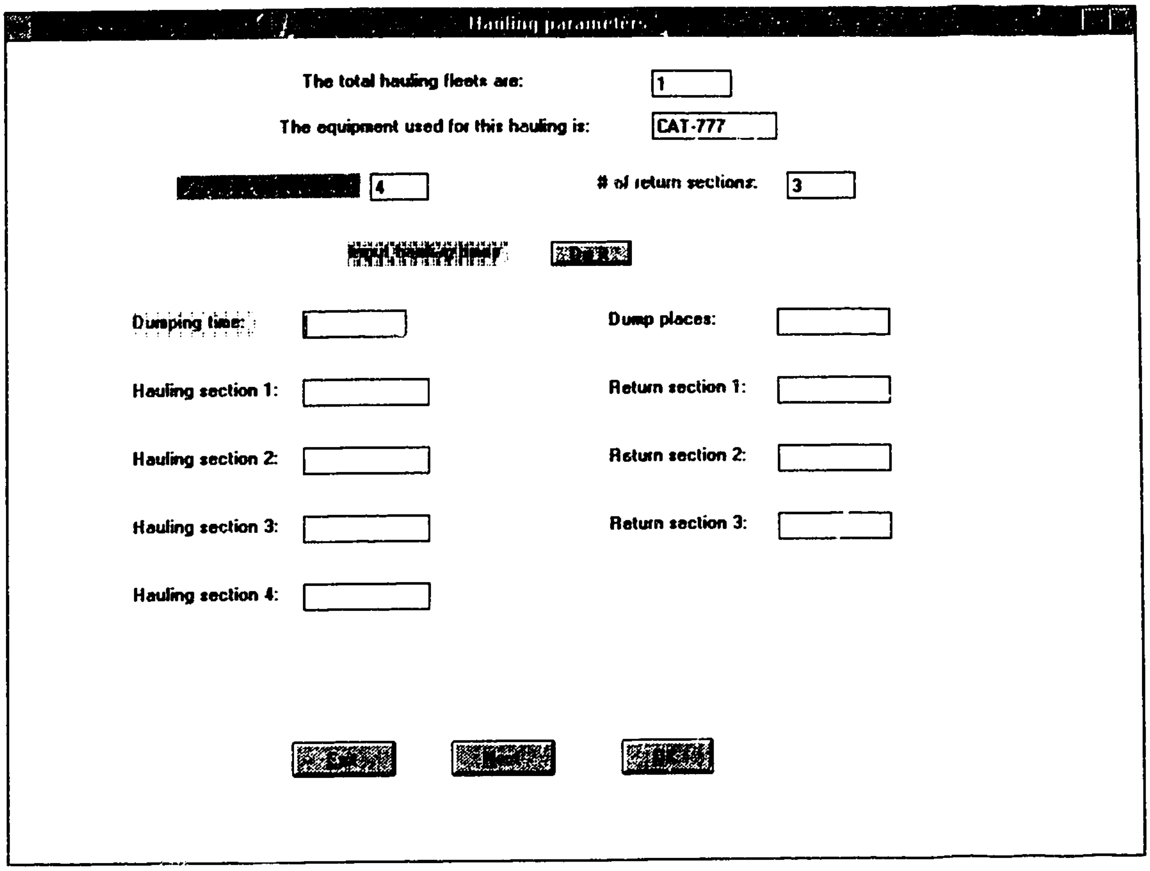
This screen is to let the user specify the activity durations involved in a bulldozing operation. The user can directly input a constant duration or a distribution.



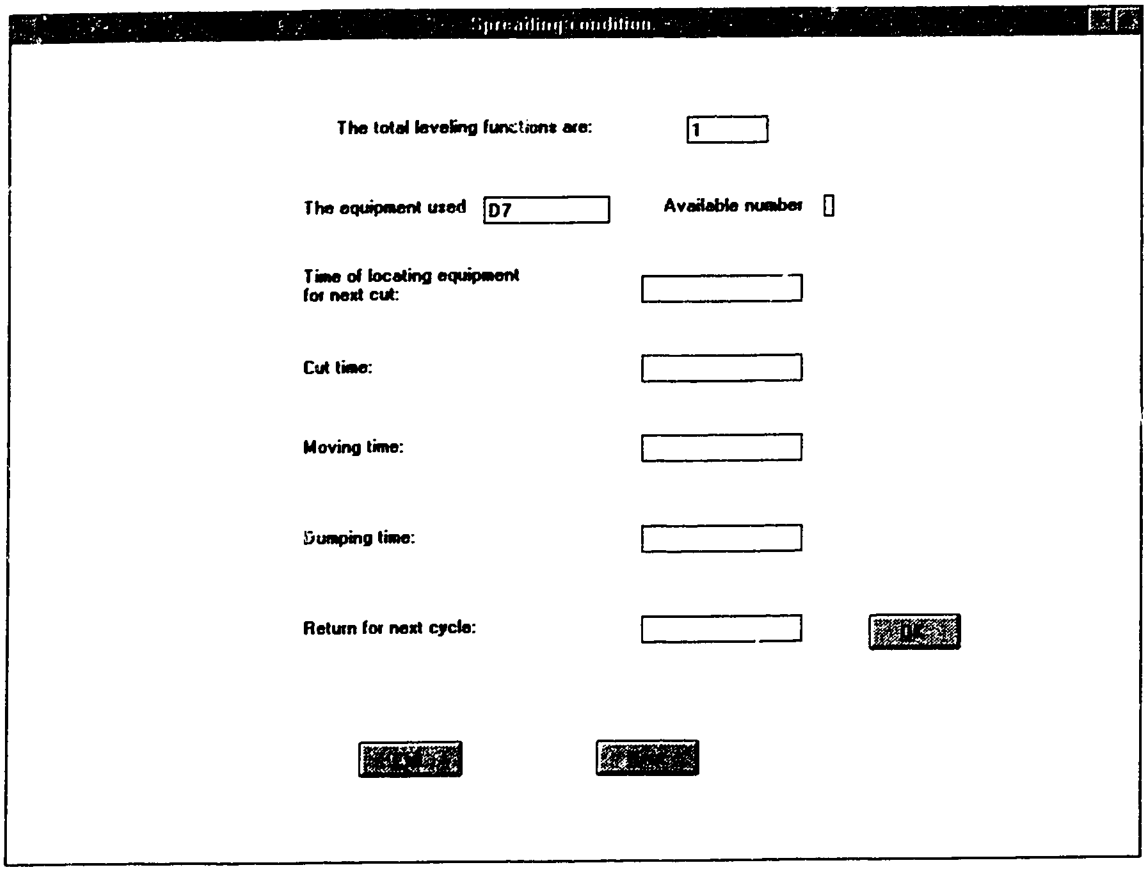
The user can use this screen to specify a distribution by giving required parameters in given boxes instead of inputting a distribution directly as described on the previous screen. This screen is available whenever the user double clicks a distribution box.



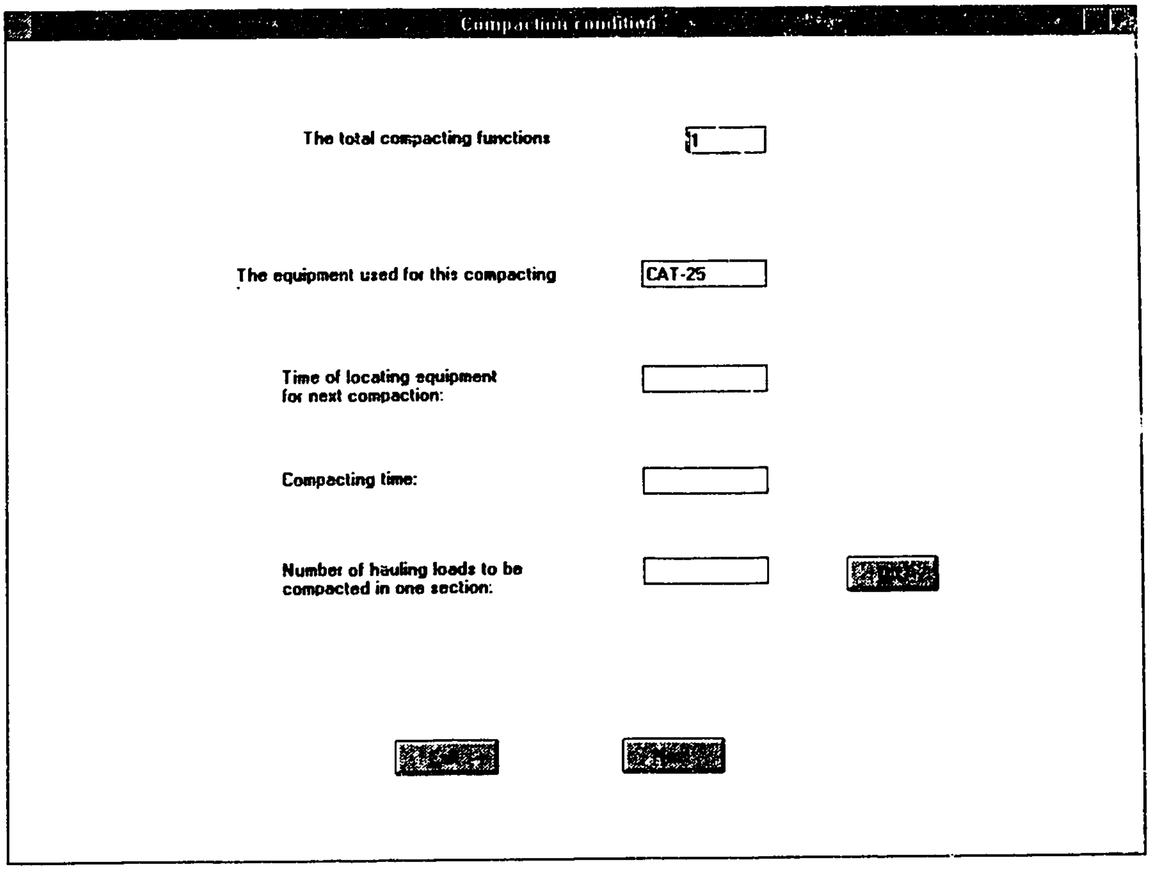
This screen is to let the user specify activity durations involved in a loading operation.



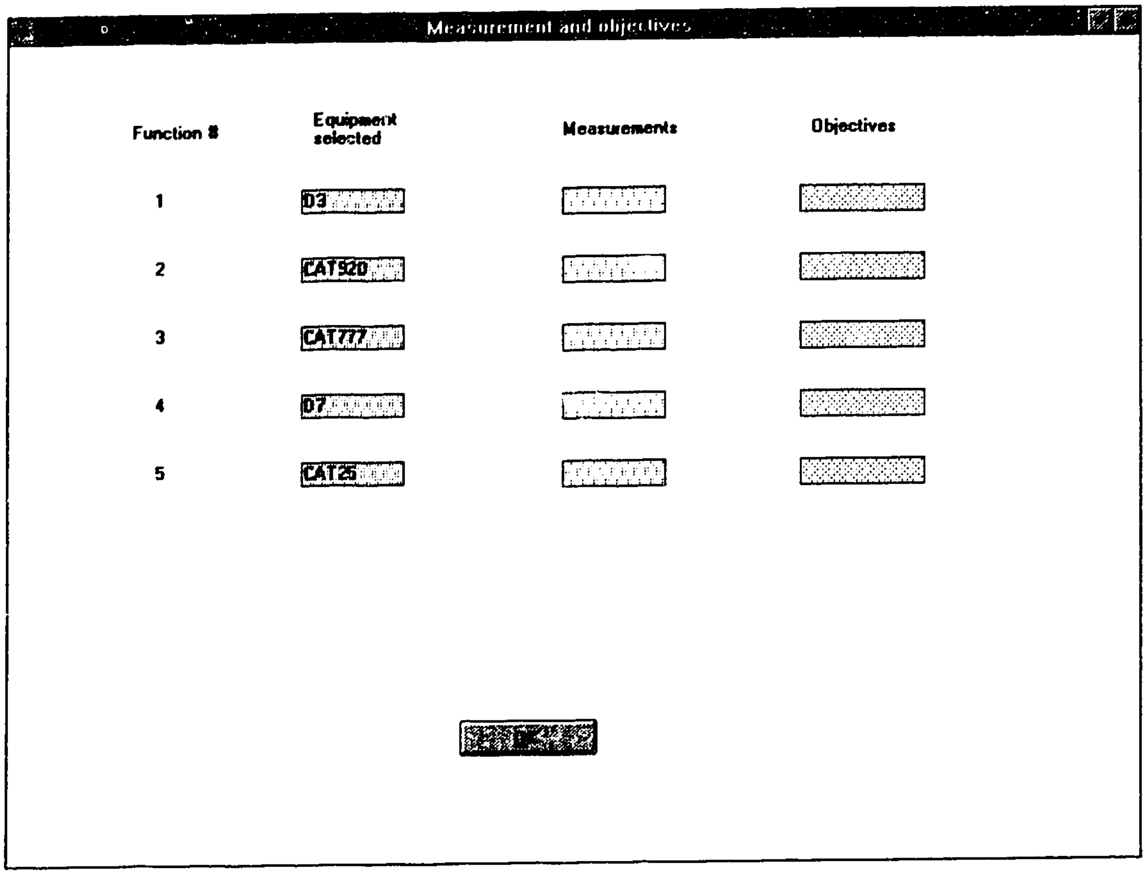
This screen is to let the user specify activity durations involved in a hauling operation.



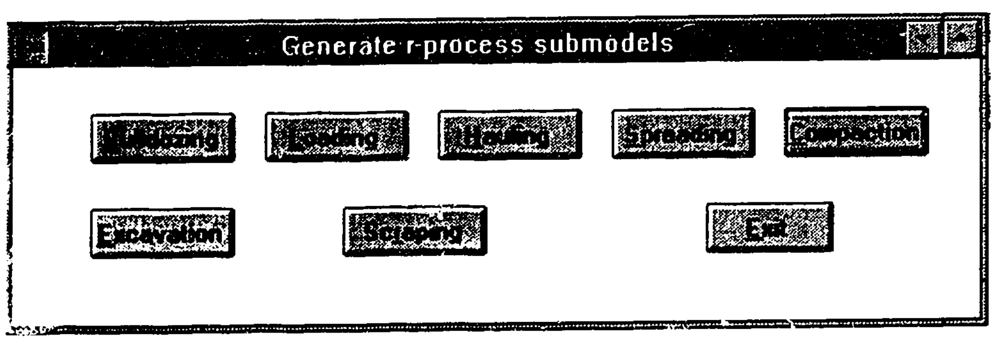
This screen is to let the user specify activity durations involved in a spreading operation.



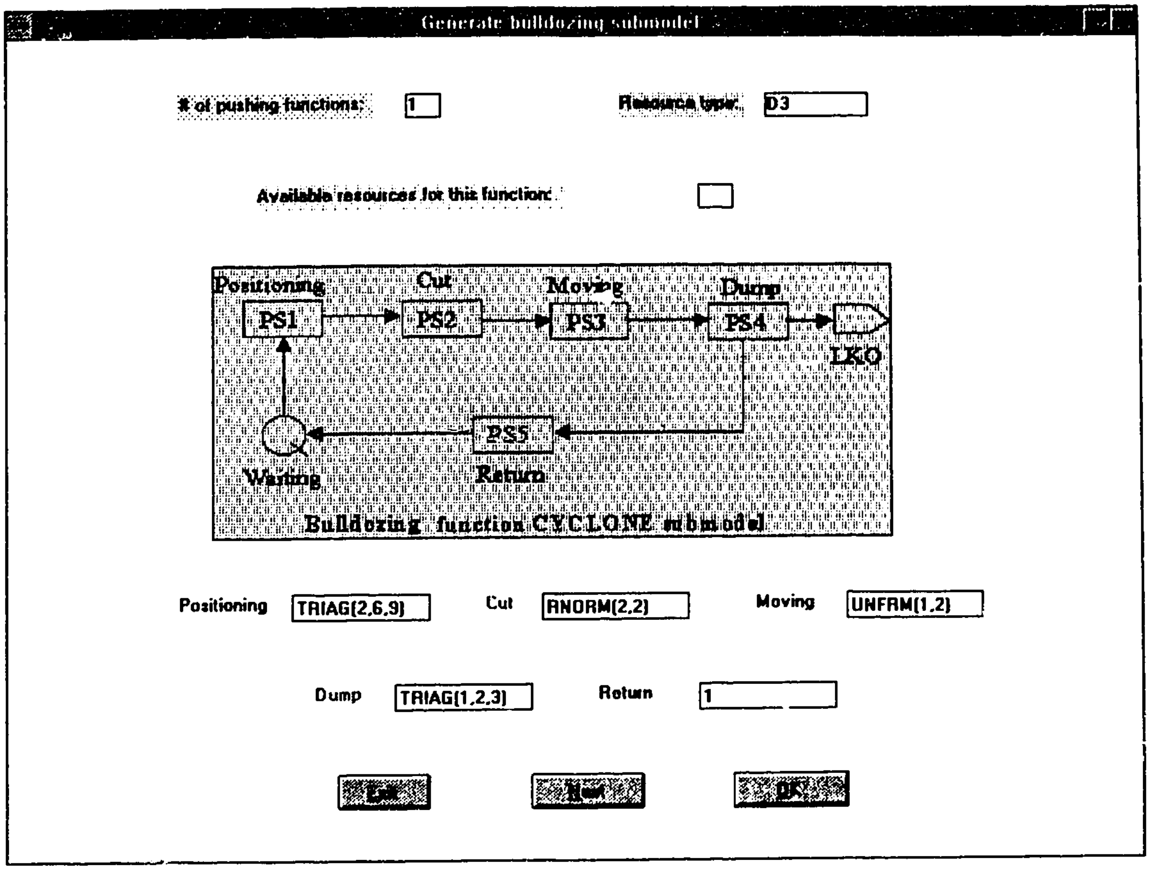
This screen is to let the user specify activity durations involved in a compaction operation.



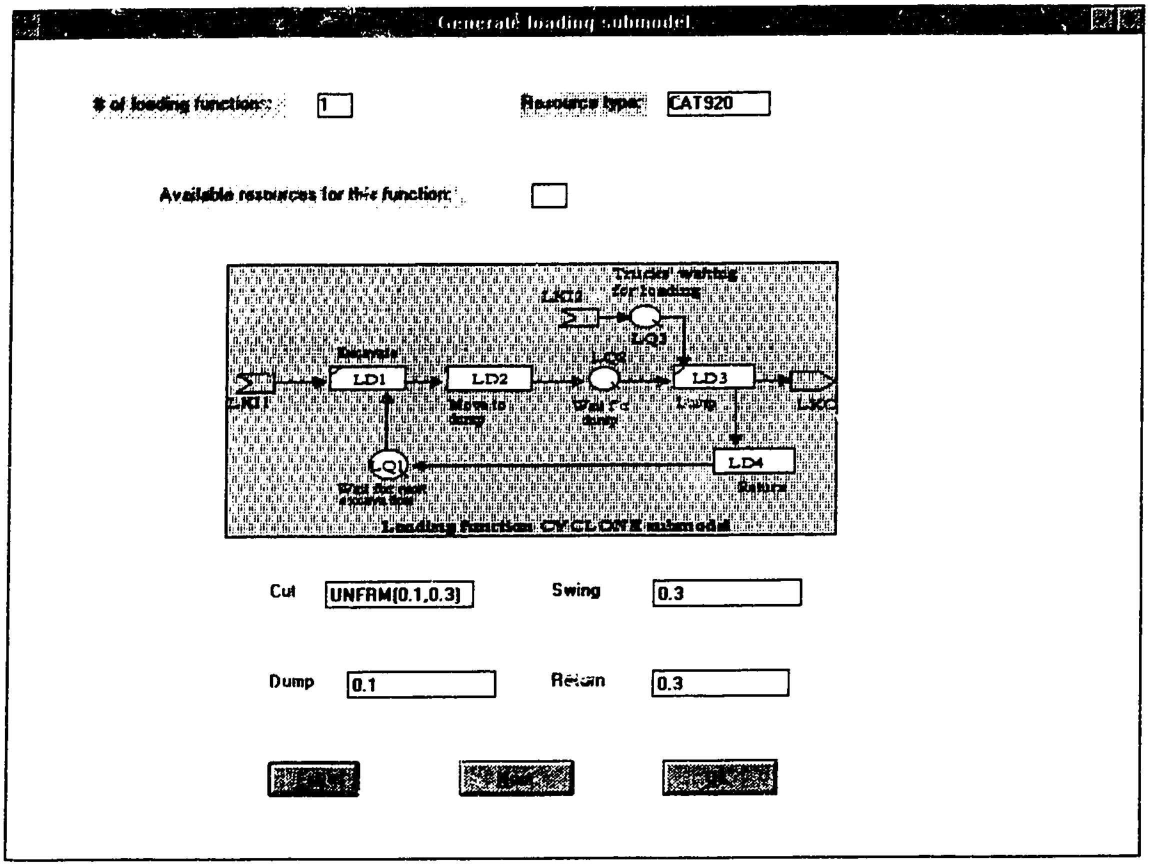
This screen is also a part of project specification. The user can specify the measurement and objective for each *r-process* through it.



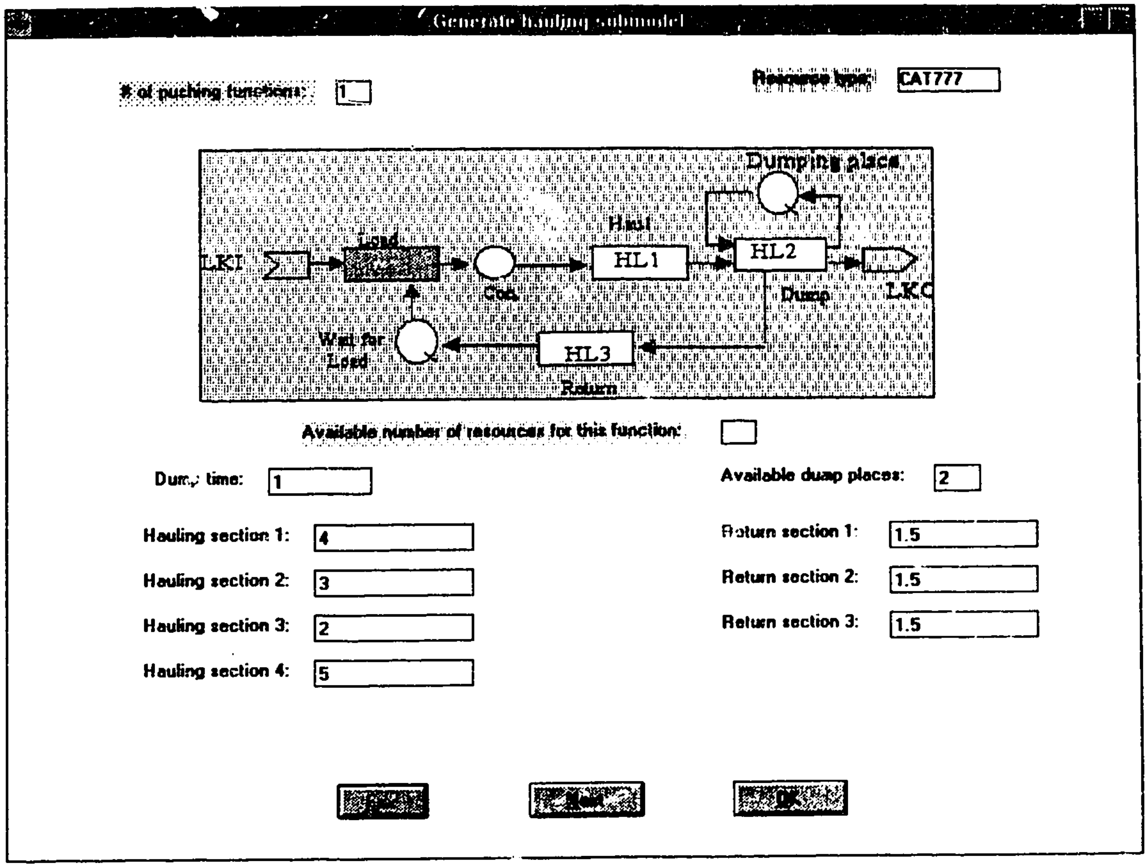
After resource and project specifications, the user can generate SLAM II simulation models which are divided into two phases: *r-process* model generation and entire model generation. Seven types of r-process models can be generated with the user clicks appropriate buttons as shown on this screen.



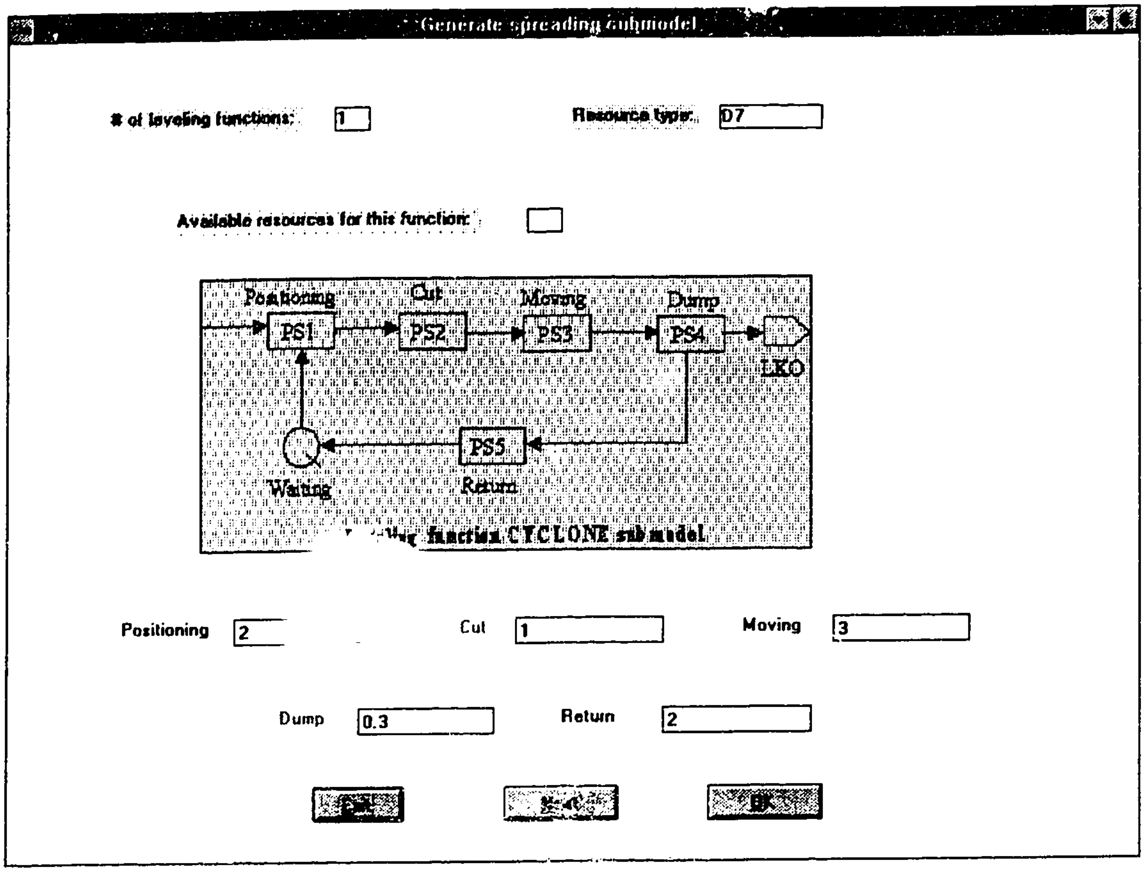
This screen allows the user to generate *r-process* models for bulldozing operations. A CYCLONE-based model is shown on the screen for the interesting user to understand the atomic model of the bulldozing operation. All user-specified parameter values are shown on this screen to let the user examine them and make the final changes. If there are multiple bulldozing *r-processes* in a project, the user should click "Next" to generate additional *r-process* models.



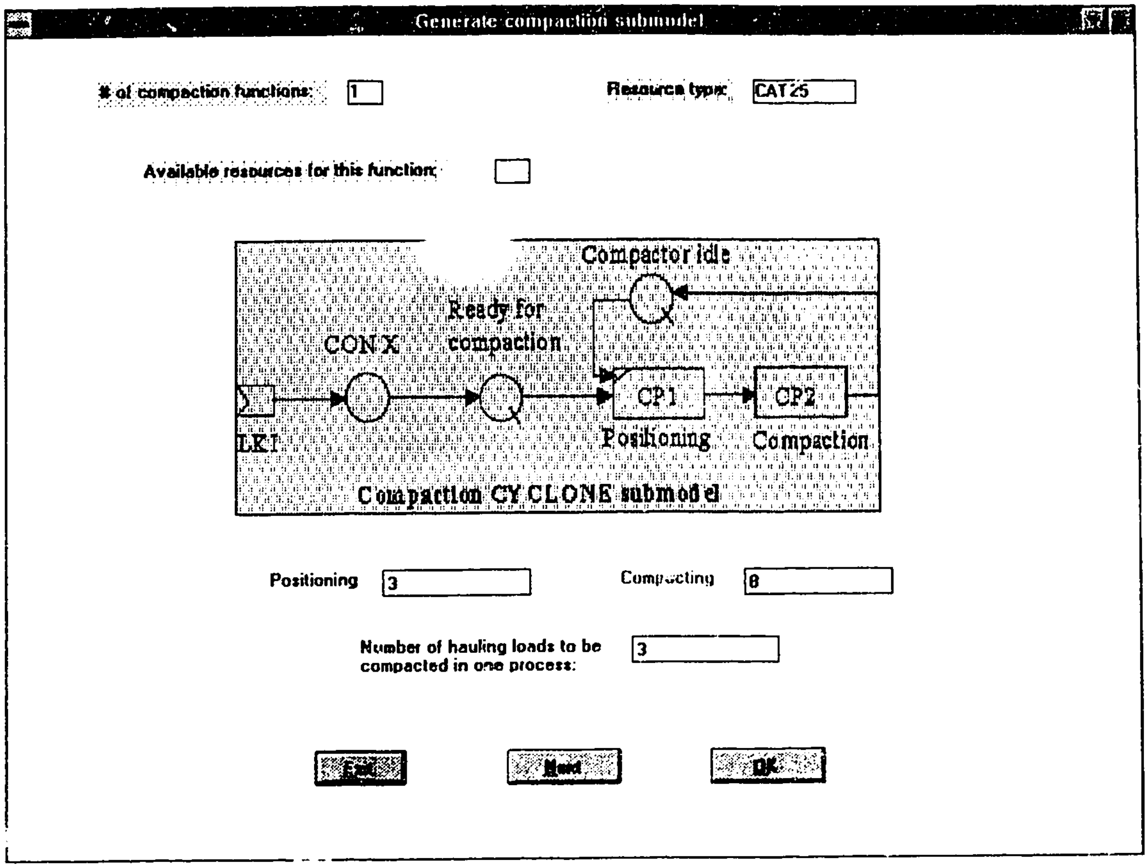
This screen allows the user to generate *r-process* models for loading operations. A CYCLONE-based model is shown on the screen for the interesting user to understand the atomic model of the loading operation. All user-specified parameter values are shown on this screen to let the user examine them and make the final changes. If there are multiple loading *r-processes* in a project, the user should click "Next" to generate additional *r-process* models.



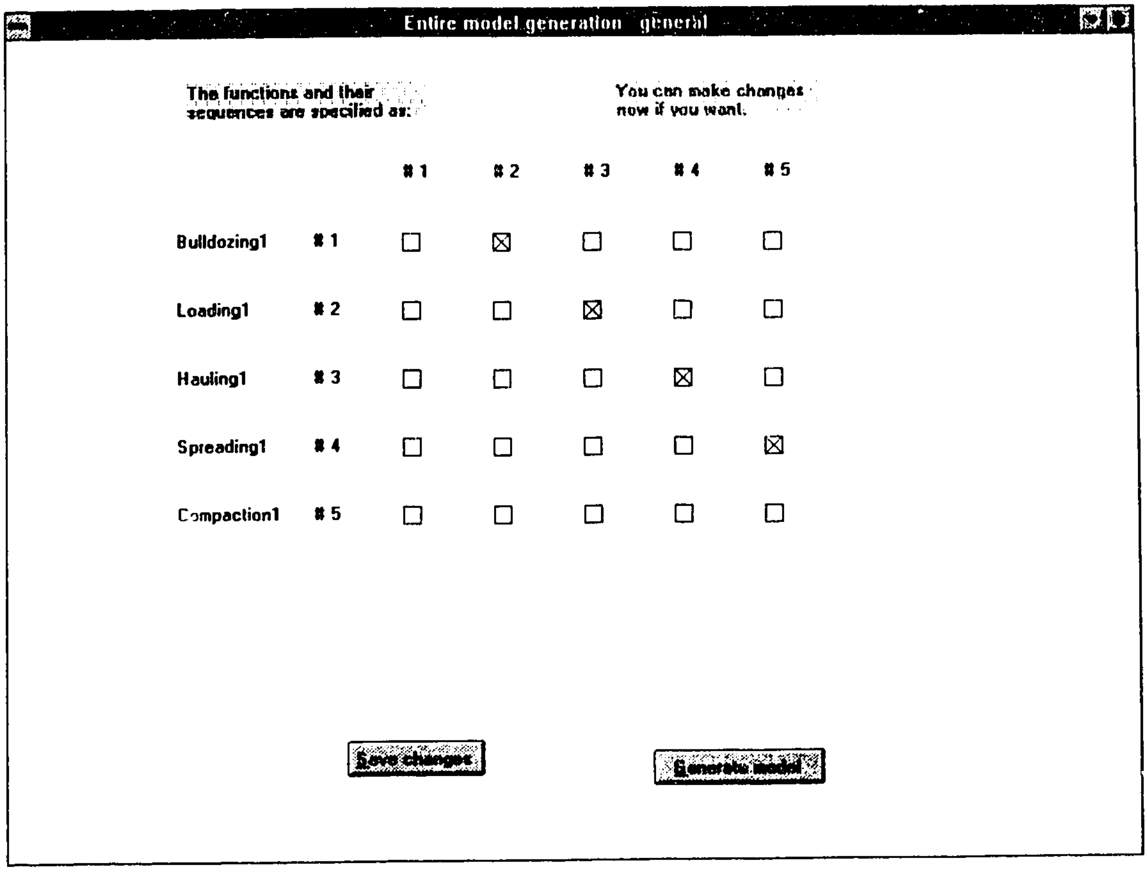
This screen allows the user to generate *r-process* models for hauling operations. A CYCLONE-based model is shown on the screen for the interesting user to understand the atomic model of the hauling operation. All user-specified parameter values are shown on this screen to let the user examine them and make the final changes. If there are multiple hauling *r-processes* in a project, the user should click "Next" to generate additional *r-process* models.



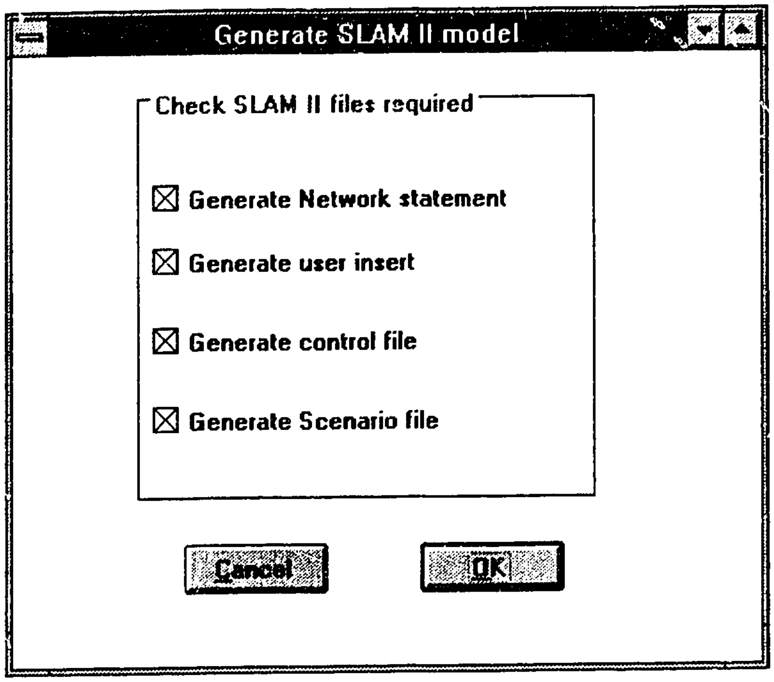
This screen allows the user to generate *r-process* models for spreading operations. A CYCLONE-based model is shown on the screen for the interesting user to understand the atomic model of the spreading operation. All user-specified parameter values are shown on this screen to let the user examine them and make the final changes. If there are multiple spreading *r-processes* in a project, the user should click "Next" to generate additional *r-process* models.



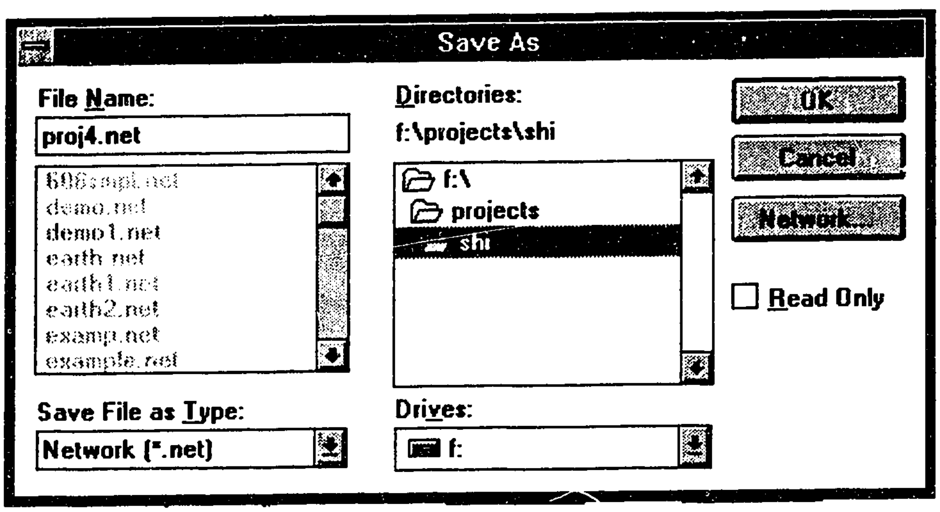
This screen allows the user to generate *r-process* models for compaction operations. A CYCLONE-based model is shown on the screen for the interesting user to understand the atomic model of the compaction operation. All user-specified parameter values are shown on this screen to let the user examine them and make the final changes. If there are multiple compaction *r-processes* in a project, the user should click "Next" to generate additional *r-process* models.



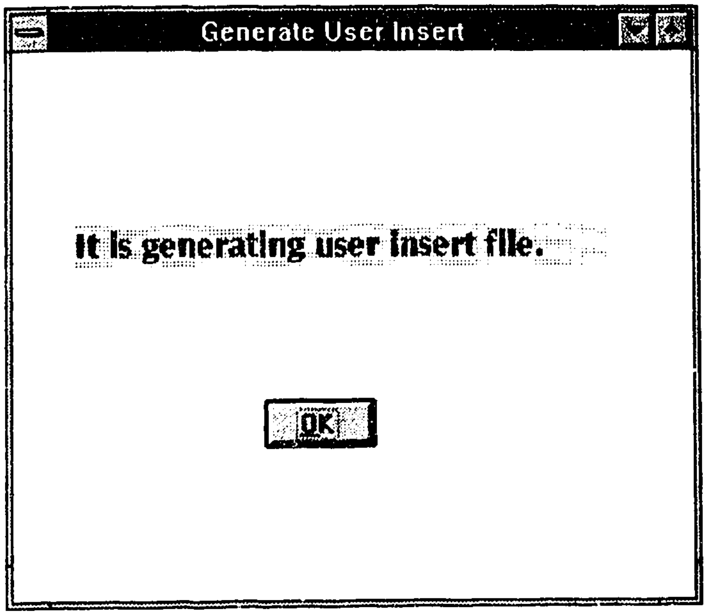
After all *r-process* models have been generated, an entire SLAM II simulation model can be constructed. This screen is activated by clicking "Entire model generation" item of the "modeling" menu showing on the main menu screen. Information shown on this screen was previously specified by the user and the user can make the final changes. After confirming all information, the user can click "Generate model" button.



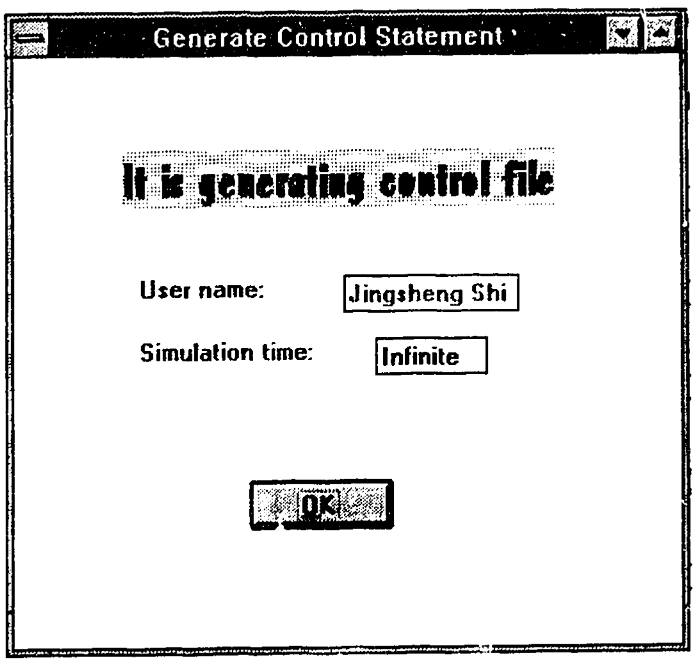
After the user clicks "Generate model" button, fo· ·les required for SLAMSYSTEM are recommended for generating. If only some of these files are expected, the user can check what he/she needs.



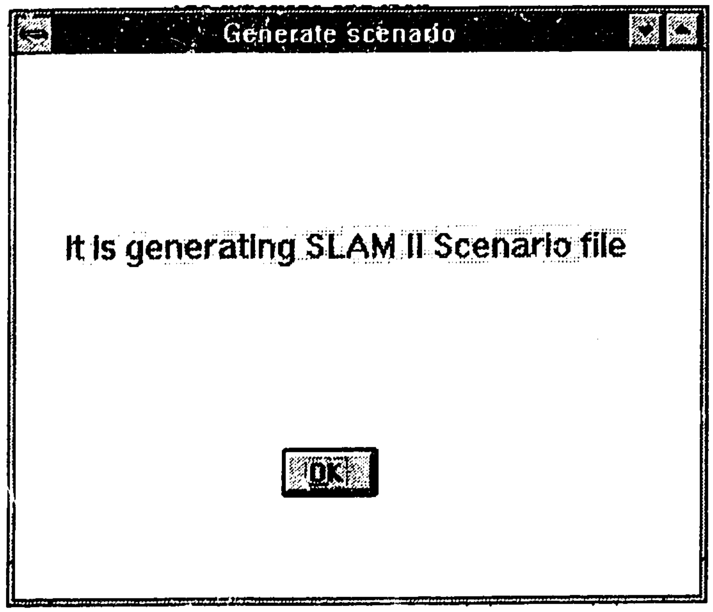
The user can save the generated files to a designated location as shown on this screen. Because SLAM II can only open an application from designated "project" directory, the generated files are recommended to be saved there.



A user insert file can be generated with the user clicking "OK".



A control file can be generated with the user clicking "OK". The user can specify the user name and expected simulation time.



A scenario file can be generated with the user clicking "OK". After generating a SLAM II model for a project, the user can save the project for later use and then switch to SLAMSYSTEM to open the generated application and simulate it.

VITA

Jingsheng Shi was born on June 19, 1962, in Hubei, P.R. of China. Upon completion of his B.Sc. degree in Hydraulic Construction at the Wuhan University of Hydraulic and Electric Engineering (WUHEE) in July 1982, he continued to pursue his graduate study in Construction Management at WUHEE and was awarded Master of Science degree in May 1985.

Since then, he was working at WUHEE as an Instructor and Lecturer. He had taught undergraduate and engineer training courses in construction management, computer applications, operations research, risk analysis and decision making. He had also conducted miscellaneous research and consulting projects in simulation, construction planning and scheduling, expert system, and mathematical modeling.

Mr. Shi was admitted to the University of Alberta and was awarded the University of Alberta Ph.D. Scholarship in September 1992 to the Ph.D. study in the Department of Civil Engineering (Construction Engineering and Management).