A CAD-Based Simulation Modeling Methodology for Construction

By

Kevin R. Mather ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment

of the requirements for the degree of Master of Science

in

Construction Engineering and Management

Department of Civil and Environmental Engineering

Edmonton, Alberta

Spring 1998

0-612-28964-8

Faculty of Graduate Studies and Research

The undersigned certify that they read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **A CAD-Based Simulation Modeling Methodology for Construction** submitted by **Kevin R. Mather** in partial fulfillment for the degree of **Master of Science in Construction Engineering and Management.**

Dr. S.M. AbouRizk

Prof. S.P. Dozzi

Dr. P.M. Steffler

Dr. M. Brett

DATE: 3/23/98

Simulation modeling of construction operations can be a useful tool for estimating, planning and controlling construction projects. Computer simulation provides advantages in modeling the dynamics of construction projects by incorporating the random distributions of events at an activity level. Computer Aided Design (CAD) programs are commonly used design platforms which contain immense amounts of information regarding the project design and the construction site. This thesis presents a method of analyzing construction operations through the use of an integrated CAD-based simulation system. The integrated system incorporates the site and design information from CAD and uses this data: (1) to define the simulation model from predefined atomic models, and (2) to link this information to simulation model activities and components.

Once the project structure and site have been defined in CAD, the simulation models can be used to analyze construction operations, different construction methods, changing site conditions and system productivity. The presented system uses an earthmoving operation as an example. The earthmoving system incorporates neural networks for excavator productivity estimation based on site conditions and soil types which are input from CAD.

The author wishes to thank his supervising professor Dr. Simaan AbouRizk for his encouragement, support and technical guidance during the course of the research. Appreciation is extended to the members of my examining committee, including Dr. Steffler, Prof. Dozzi, and Dr. Brett.

CHAPTER 1

THESIS SCOPE AND OBJECTIVES

1.1 INTRODUCTION

Construction projects are unique one-time undertakings with only one attempt to execute a successful project. Construction management of these projects involves planning, scheduling and controlling projects to meet schedule and financial goals. Construction managers must use the tools and resources available to them to successfully plan and execute a project. To assist them in meeting project goals, construction engineers use many analytical techniques, as well as intuition and previous experience. These factors play important roles in the construction manager's decision-making process.

Traditional methods of project planning and control techniques include Critical Path Method (CPM), Program Evaluation Review Technique (PERT), linear scheduling, and bar charts. CPM schedules use a network approach to determine early start, late start, early completion, and late completion dates of activities in a project. PERT is used to quantify the impact of variation within the project by describing activity durations with random distributions. Bar charts are the predominant scheduling method in the construction industry. They graphically plot activities versus time and are widely accepted due to their simplicity. Linear scheduling methods are used to describe projects that are repetitive in nature. They plot lines of constant or varying slopes to represent activities, with the Y-axis representing the repetitive units and the X-axis representing time.

dynamic interactions between resources and the processes that utilize them (AbouRizk and Shi 1994, Paulson et al. 1987). While the described tools are useful for planning, they lack consideration of this inherent randomness and fail to address the dynamic interactions between resources and processes involved in construction.

Computer simulation is one method that is used to quantify the variability and dynamic interactions between resources within construction activities. Computer simulation provides advantages in quantifying the dynamics of construction projects by representing the durations of tasks with random distributions and by modeling the interactions between resources. Simulation is used for design, procedural analysis and performance assessment of a modeled system. In construction, simulation can be used to evaluate expected project costs, project and activity durations, production rates, resource utilization, risk analysis, and site planning.

Simulation systems, although effective, have not been accepted into the construction industry. Accurate simulation of construction projects requires large, complex models. The development and testing of these models is extremely time consuming and requires intimate knowledge of both simulation and of the system being modeled. It has been shown that the time and effort required to build a simulation model for a construction project does not provide a positive return on investment (Shi and AbouRizk 1994).

From the perspective of a construction manager an ideal simulation system would be fully automated, require minimal simulation knowledge, be simple to use, provide quick results, and add a dimension of accuracy to planning and control that existing analysis tools do not possess. In order to be accepted in the construction industry, the simulation system must be presented in a very simple and graphical context (Shi and AbouRizk 1994). In the construction business, most contractors specialize in a specific area of expertise; therefore they are logically only interested in simulation tools that are specific to their business. Special purpose simulation tools can be developed for a specific area of construction. These tools can be presented in a simple and graphical context, require minimal simulation background, and provide quick results (AbouRizk and Hajjar 1997).

Other powerful tools that can be used by managers are Computer Aided Design (CAD) and Computer Aided Design and Drafting (CADD) systems such as AutoCAD and MicroStation. Construction managers seldom use these tools; this is due in part to the complex nature of the CAD environment, and the lack of construction-specific CAD tools. Although seldom used by construction managers, these tools are considered essential in today's engineering and design industry. Since design using CAD systems is so common, and there is an immense amount of information regarding project design and site specifics contained in CAD files, construction project control tools using CAD integration can provide numerous benefits to users.

The author proposes that by integrating CAD and simulation, analysis tools can be created to harness both the site and project design information from CAD and the analysis powers of simulation.

The existing state of integration between CAD and simulation systems is dependent on the context of the term 'simulation'. According to Pritsker (1997), simulation is "... the process of designing a mathematical-logical model of a real system has been in the form of animation. Animation as defined by the Oxford English Dictionary (1985) is "... a technique of photographing successive drawings or positions of puppets to create the illusion of movement." The majority of the work to date in the integration of CAD and simulation is merely animation. This thesis presents an approach that extends the integration of CAD and discrete event simulation thus truly reaching an integrated model.

1.2 SCOPE OF RESEARCH

1.2.1 Introduction

The current level of integration of CAD with simulation analysis tools has, for the most part, been limited to animation. The scope of this research focuses on the integration of CAD and simulation for the purpose of analyzing construction operations. The need for a system that integrates CAD and simulation has been put forth by both researchers and construction simulation practitioners. The modeling and integration methodology presented in this research can be used as a framework for the creation of special-purpose, CAD-based simulation modeling tools. A prototype CAD-based simulation modeling system will be implemented using an earthmoving operation for illustrational purposes.

CAD design files can contain tremendous amounts of information regarding the design of the structure to be constructed and the surrounding site conditions. This information can be used to improve the planning and control of construction operations directly from CAD. The integration of CAD and simulation allows CAD-based data to have the potential to more accurately depict construction processes because they can accurately depict the conditions that will be encountered. By automating the transfer of CAD-based simulation specific information, the proposed system minimizes required data entry, thus simplifying the simulation process.

The prototype CAD-based simulation modeling earthmoving system illustrated throughout this thesis is designed to be a fully functional earthmoving simulation system. The system is designed to be practical for use by construction practitioners, simple to use, require minimal simulation knowledge, provide quick results, and add a level of accuracy beyond the analysis methods that are currently used for earthmoving projects.

As part of the earthmoving work, a method of determining excavator productivity as a function of the soil conditions extracted from CAD is presented. This portion of the research accentuates the importance of the integration of the CAD data.

Before proceeding into the main body of the thesis, it is important to develop an understanding of the entire scope of this research. The research prototype used is an earthmoving operation. It will be used for examples throughout the entire thesis.

Earthmoving operations act as effective examples for simulation research because they are easy to visualize and it is easy to follow the construction logic involved. Earthmoving projects indicative of the system described herein include dam construction, overburden removal, and surface mining. These projects are costly operations and are very equipment intensive.

distinct components, as illustrated by Figure 1-1:

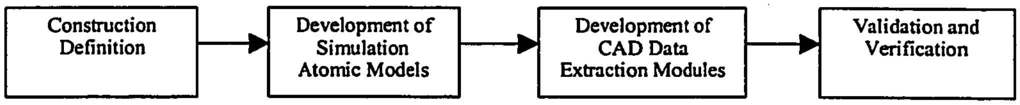


Figure 1-1 Development of CAD-Based Simulation System

1) The first component of development is the construction definition. The construction processes, the construction site and the critical factors affecting the construction processes must be identified. The locations of the information defining these factors should be determined so that they can be utilized in the simulation.

2) The second component is the creation of 'atomic' simulation models representing the construction activities. Atomic models are basic components that allow development of full models dynamically during a simulation run by integrating required components. These atomic models mimic the construction activities and are combined to create a system model. The system model uses the information from CAD and simulates the construction.

3) The third component involves relating the information available in CAD to the simulation models. The CAD data is extracted using CAD extraction modules, which are developed specifically to relate to the simulation atomic models. These extraction modules can be used to define general site information, specific site data, construction sequencing, construction resources, and construction processes for the project structure.

CAD-based simulation system.

1.2.2 Construction Definition

The first step in the development of a CAD-based simulation modeling system is to define the construction operation. The construction definition consists of a description of the site, the project structure, and the construction process. For the purpose of earthmoving, there are three major components in the site and project structure, as illustrated in Figure 1.2.

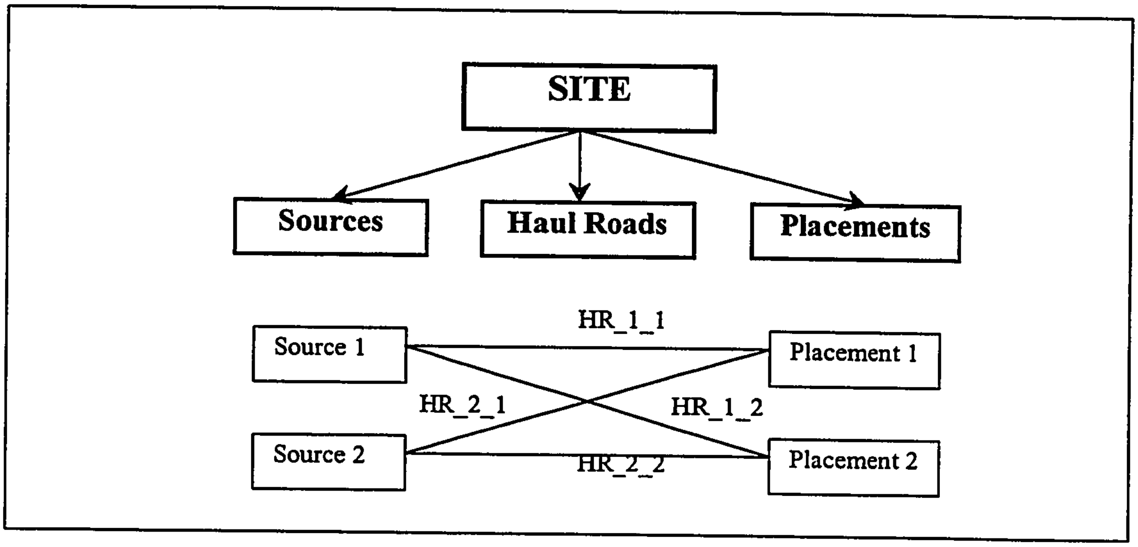


Figure 1-2 General Site Description

The source is the location at which the excavating will occur. The haul roads are used to transport the material via trucks from the source to the placement (the area at which the material will be dumped). A haul road is defined as a route from a source to a placement, thus with two sources and two placements, there are 4 possible different haul routes. For example, the haul road from Source 1 to Placement 1 is described by HR_1_1. The example for this thesis will use a single source (or excavation) and a single placement location connected with one haul road, as described in Figure 1-3.

Earthmoving is a cyclic process as illustrated in Figure 1-3 and consists of five basic tasks:

1) Excavation and loading
2) Travel-loaded (from the source to the placement)
3) Dumping
4) Placing (dozing, grading and compacting)
5) Travel-empty (returning from the placement to the source)

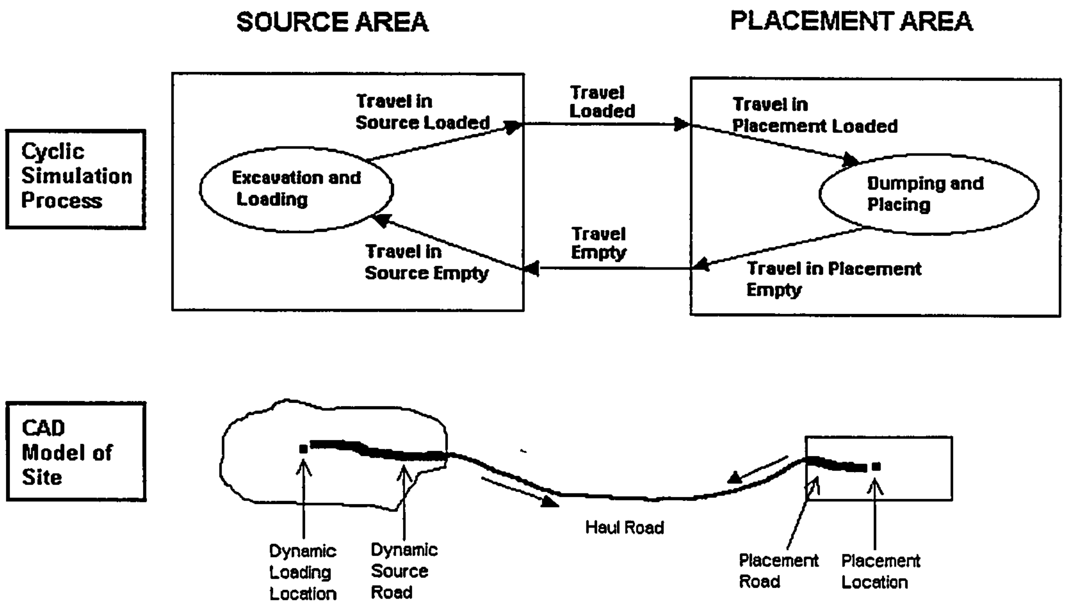


Figure 1-3 General Earthmoving Description

process at a high-level, but to evaluate different construction methods and sequences, it must be broken down into greater detail. This work focuses on the detailed definition of the excavation location. The excavation is detailed to a level enabling the construction process to be clearly visualized. Although no definition of the placement location is used, the procedure used in the excavation could be used to define the placement.

An excavation of any significant depth requires the excavation to be performed in benches as illustrated in Figure 1-4. The depth of each bench and the number of benches is determined by the total excavation depth as well as the size and type of the excavator.

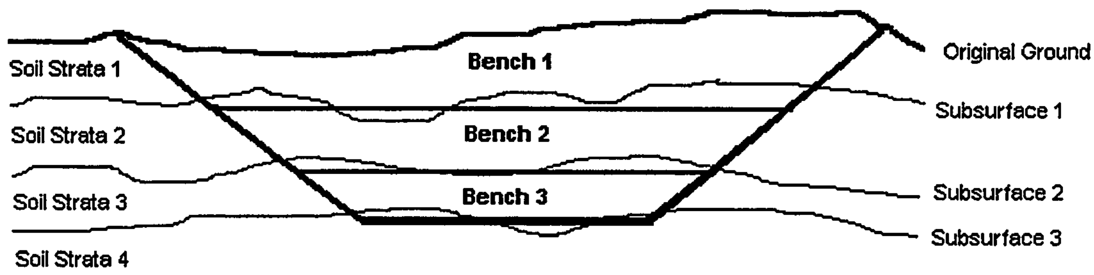


Figure 1-4 Excavation Typical Cross Section

After the excavation has been divided into benches, it can be further defined by dividing the plan view into '**subgrids**' as illustrated in Figure 1-5. This allows the sequence in which the excavation will take place to be defined. For example, the excavation may move from 'subgrid 1,1' to 'subgrid 2,1' followed by 'subgrid 3,1' and so on, until bench 1 has been completed, at which time excavation of bench 2 begins. This construction sequencing is important because as the location of the excavator moves, the distance that trucks must travel to exit the source changes. As the excavation moves within a site, the soil composition will also change. Excavator productivity is compositions will cause the excavator productivity to change as the excavation progresses.

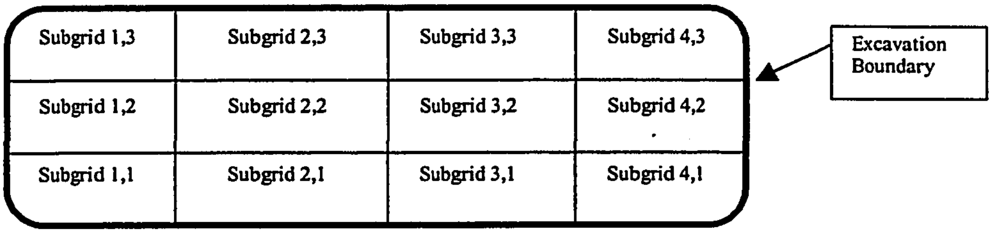


Figure 1-5 Plan View of Excavation

The excavation has now been divided in all three planes to create 3 dimensional '**blocks**'. The excavation shown in Figure 1-4 and Figure 1-5 consists of 3 benches, each containing 12 subgrids or 36 different excavation blocks. This breakdown of the excavation makes it possible to define the construction sequence of these blocks. The type of loading operation can also be defined for individual blocks. The types of loading equipment to be used for different blocks can also be defined. Each block possesses its own individual properties as described in Figure 1-6.

Each block possesses physical properties such as total volume, soil composition, and average depth. Since a CAD earthmoving file contains information on soil layers, the physical properties of each block can be determined. From the discussions on the excavation definition, and earthmoving operation, the construction process is defined, and the simulation models can be developed.

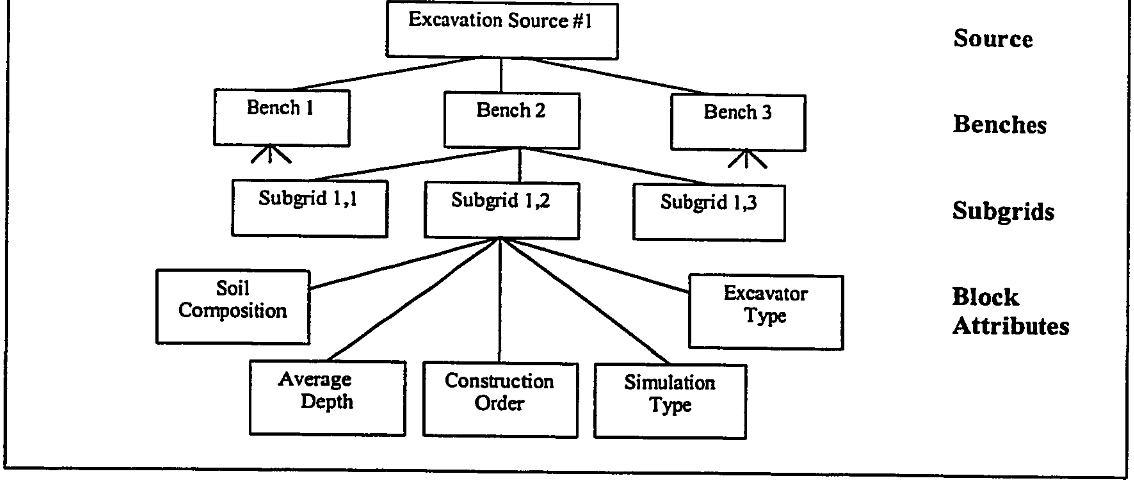


Figure 1-6 Excavation 'Block' Properties

1.2.3 Simulation

Simulation in construction has been used successfully in academic research but has had limited success in industry due to the complexities involved in constructing and testing models (Shi and AbouRizk 1994). The accurate modeling of construction operations requires large complex models that are difficult to develop and validate. The need to simplify models has been identified as a primary subject that simulation modeling must address (Ibbs 1987). Zeigler (1987) and Luna (1992) have presented hierarchical and modular modeling concepts as a tool for simulation simplification.

The basic components of the modularity concept include the 'atomic model', the 'model base' or 'model library', and 'coupling'. An atomic model can be defined as a basic and unique description of a particular process. The 'model library' consists of a collection of atomic models, which combine to construct a high-level model. Coupling is the act of combining the related models. Developing specific atomic models and using to be developed as illustrated in Figure 1-7. The combined use of modular concepts and CAD integration presents a method of simplifying modeling.

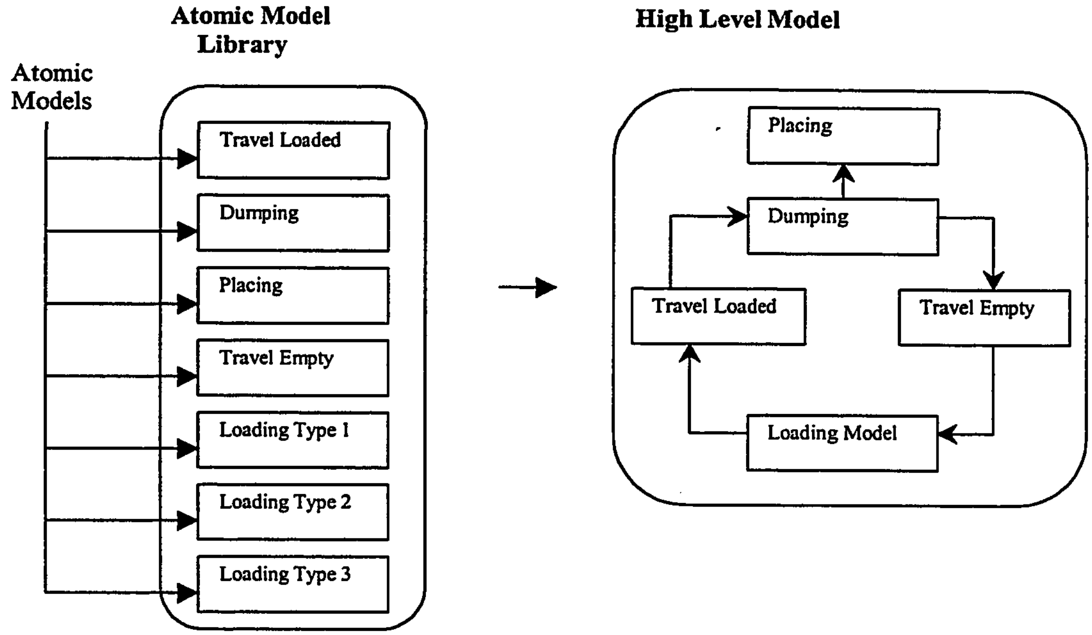


Figure 1-7 Modular System for Earthmoving Operations

Since the productivity of the system is often constrained by the loading operation, the need arises to model this operation at a detailed level. Six different methods of excavating and loading have been identified: three for shovels and three for backhoes:

1) Shovel and trucks: Loading both sides

2) Shovel and trucks: Loading one side

3) Shovel and trucks: Loading drive-by

4) Backhoe and trucks: Top loading, back-in

5) Backhoe and trucks: Top loading, drive-by

Each loading operation is characterized by different interactions between resources and is represented by an individual atomic model. The resulting productivity of the loading operation is affected by these interactions and is quantified by the simulation.

1.2.4 Computer Aided Design

Computer Aided Design (CAD) programs are commonly used for design but are seldom used for the construction phase of a project. The design files contain immense amounts of data regarding the project design and the site specifics. The vast design information stored within CAD files creates a logic that simulation integration with CAD can provide real time savings in defining the simulation system model and for extracting data used by the simulation models. Through integration with the CAD design files, the simulation can harness the benefit of the site and design information from the CAD environment. Figure 1-8 provides a high-level overview describing the CAD-based simulation methodology.

In the CAD-based simulation modeling methodology, CAD and simulation are linked to form an integrated system. The links are created from CAD and are termed CAD extraction modules. The simulation and the CAD structures are maintained as separate entities connected only through an intelligent data manager. By maintaining individuality, the required level of simulation knowledge and interaction with the simulation module can be minimized, thus simplifying simulation modeling.

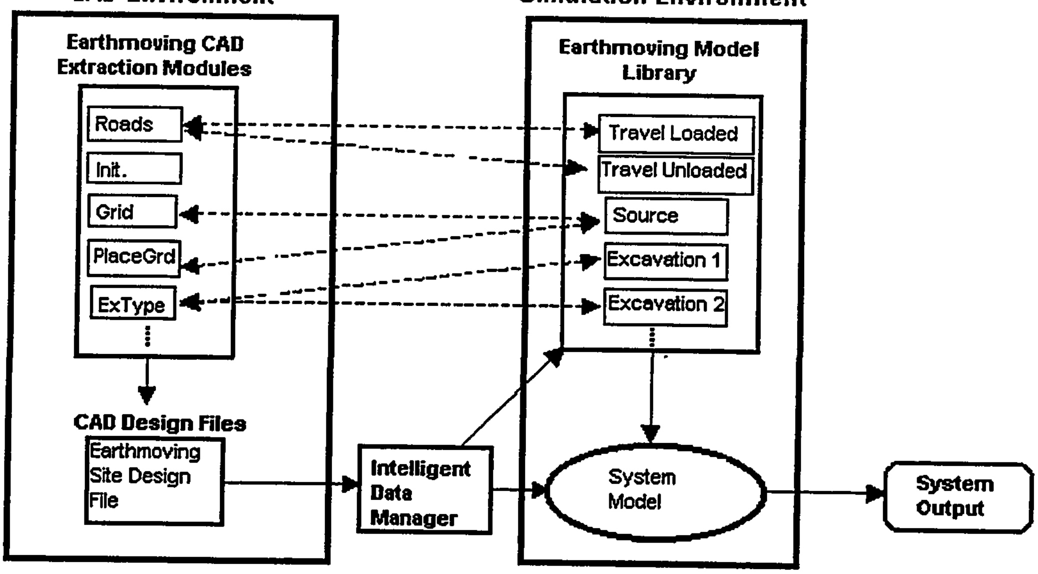


Figure 1-8 CAD-Based Simulation Modeling Methodology

The CAD-simulation linking is performed using CAD extraction modules. These modules are used to define general site information, specific site data, construction sequencing, construction resources, and construction processes for project elements. The CAD extraction modules are used (1) to extract data and relate this information to atomic models, and (2) to define the high-level system model by combining atomic models.

1.2.5 Excavator Productivity Estimation

As mentioned in Section 1.2.2, the productivity of an excavator varies with soil composition and site conditions. A need to quantify this variation was identified by a local earthmoving contractor. The method of productivity estimation used for this research is a form of artificial intelligence called neural networks. The proposed neural excavator and truck relative positioning, and excavator footing conditions.

1.2.6 Integration

The integration of CAD, simulation atomic models, and neural networks forms a complete special purpose simulation system that can be used to analyze earthmoving operations. The system is presented below in Figure 1-9.

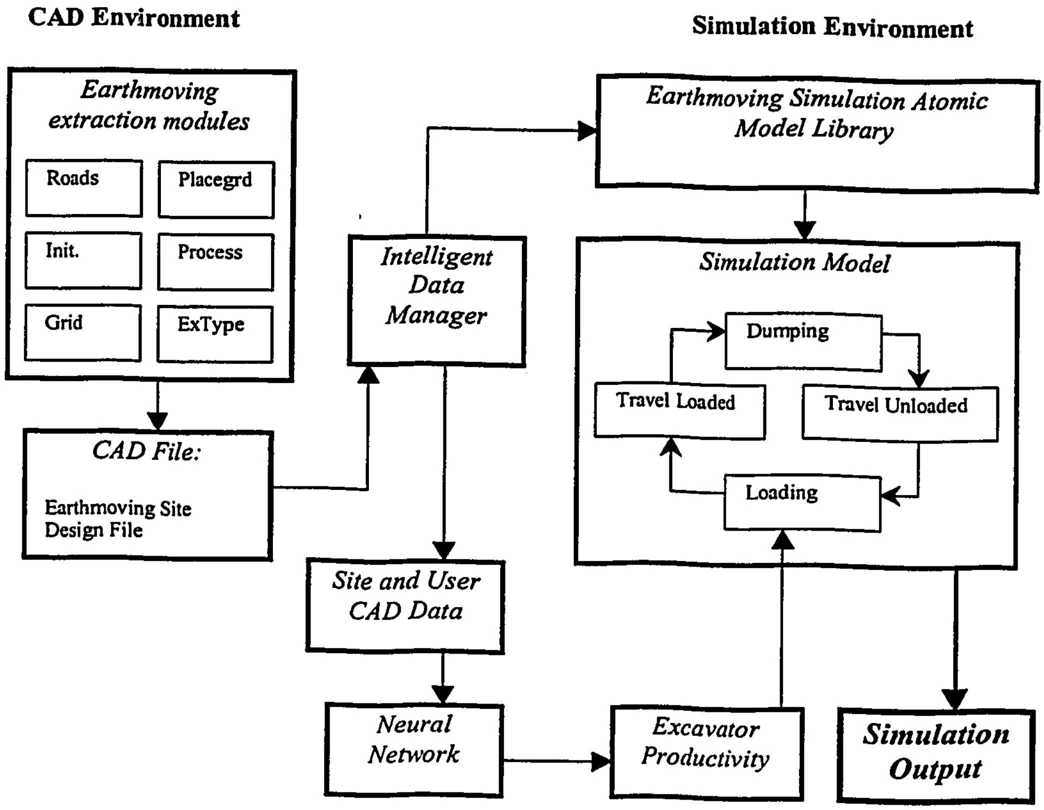


Figure 1-9 Integrated Earthmoving System

The primary objective of this research is the development of a CAD-based simulation methodology for construction operations.

The secondary objective of this research is the development of a special purpose CAD-based simulation model for earthmoving using modular simulation methods for model creation, with a specialized focus on the excavation process.

The tertiary objective of this research is to introduce a method for using neural networks for excavator productivity prediction.

The CAD-based simulation modeling methodology has the potential to provide many benefits. The first benefit involves improving the accuracy of the simulation through the incorporation of site specific information from CAD. By incorporating site specific and project structure information, the project can be more accurately described for the simulation. The second benefit that the system achieves is simplification of the simulation process through automated and semi-automated data entry from CAD. The final benefit that the system can provide is simplifying the modeling environment by using modular concepts to allow users to define the high-level simulation models from CAD.

1.4 THESIS ORGANIZATION

This thesis consists of seven chapters. Chapter 2 gives a comprehensive overview of the state-of-the art in simulation, CAD-simulation integration, and neural networks used for construction productivity prediction. Chapter 3 presents the use of modular simulation concepts for earthmoving, focusing on the excavation and loading operation.

research. Chapter 5 describes the use of neural networks for excavator productivity estimation. Chapter 6 incorporates the work in Chapters 3, 4, and 5; it presents the development and testing of a prototype special purpose CAD-based simulation model for earthmoving. Chapter 7 summarizes the research presented in this thesis with conclusions and recommendations for further research.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The construction industry is facing enormous technological challenges as it prepares to enter the 21st century. Increased personal computer use along with the availability of special purpose estimating, scheduling, analysis, design, and drafting tools has significantly increased the construction engineer's control abilities. Simulation tools and CAD programs are two tools that have become readily available. The ability to harness their capabilities to assist in project control functions can be a tremendous asset to construction managers. As CAD and simulation tools are further developed and become user friendly, their acceptance into the construction industry may increase.

The purpose of this literature review is to identify the current state of knowledge about the primary tools used in this thesis. Four topics are discussed in this review: First, the current level of simulation knowledge is reviewed. Second, previous works in CAD integration for construction are discussed. Third, previously developed neural network applications for the purpose of predicting construction productivity are identified. Finally, research involving earthmoving operations is investigated.

2.2 SIMULATION

Simulation is defined as the process of designing a mathematical-logical model of a real system and experimenting with this model on a computer (Pritsker 1997). The model is a description of a system, built by a user to represent the system in a mathematical manner. The dynamic nature of the system is modeled by simulation entities, which move through the simulation model during experimentation. Changes in the state of a system can occur continuously over time or at discrete instants in time. They are described by continuous modeling and discrete event modeling (Pritsker 1997). In discrete event simulation, the dependent variables in the model change at specified points in simulated time. In continuous simulation, the dependent variables of the model change continuously over time. Simulation through experimentation is used to quantify the dynamic behavior the modeled system possesses.

In general, construction simulation research has been focused in two areas: simulation language development and simulation modeling simplification. Relatively little work has focused on the need to develop simulation models at the project specific level to the point where they are useful in practice to improve on existing estimating and planning techniques.

2.2.1 Simulation Systems

AbouRizk et. al. (1992) presented the state of the art in construction simulation, focusing on the simulation systems that have led researchers to the current state. Halpin (1973) developed the CYCLONE modeling simulation methodology. A great deal of the simulation research carried out in the area of construction has been due, in part, to the success of CYCLONE. CYCLONE was the basis for a number of construction simulation systems including RESQUE (Chang 1987), COOPS (Liu and Ioanna 1992), DISCO (Haung et. al. 1994), and STROBOSCOPE (Martinez and Ioannou 1994). The methods.

2.2.2 Simulation Simplification

Simulation has not been accepted in the construction industry, in part because of the need of the user to be proficient at both simulation and in the construction operation to be modeled. Major efforts have been focused on simplifying construction modeling and automating model generation.

A number of different approaches have been used to address the issue of simplification of the modeling process. The principal approaches in this direction have been:

1) model reusability approaches
2) hierarchical and modular modeling approaches
3) computer-aided modeling approaches

The model reusability approach has centered on simplifying the modeling process by reusing previously created models for new situations. Bortscheller and Saulnier (1992) explore the possibility of re-using existing models.

Zeigler (1987) and Luna (1992) present hierarchical and modular modeling concepts. These approaches focus on the idea that combining smaller models through input and output ports can create system models. The resulting system model is a result of the coupling of smaller "atomic" models.

Computer-aided modeling approaches are based on the idea that models can be created using highly structured descriptive user input. Paul (1992) attempts to construct a simulation model where the user specifies a real world system in either natural language or activity paths.

Work has also been performed in automated generation of simulation models for construction. Shi (1995) used a resource-based modeling approach to develop a model for automated generation of models for earthmoving operations. Manavazhi (1997) developed a configuration-based modeling methodology for automation of models using high-level user descriptions.

This thesis addresses an issue put forth by previous researchers and construction simulation practitioners. Research works by Manavazhi (1997) and Shi (1995) identify the need for integration of CAD with simulation in order to incorporate design and site specific information. This research work utilizes a modular modeling approach to simplify the creation of system models.

2.3 CAD INTEGRATION FOR CONSTRUCTION

Today's complex projects require Computer Aided Design systems for design, analysis, and visualization purposes. CAD systems are used almost exclusively before the construction begins. However, this trend is starting to change. CAD is a valuable tool for construction engineers and can be used as technical support for construction operations. Using CAD to automate existing processes and provide new information for construction has several benefits. These include increased timeliness and accuracy of drawings, improved communication of technical information, and increased field productivity (Mahoney et. al. 1994).

Choi and Ibbs (1990) surveyed Construction Industry Institute (CII) members to determine trends and current status of computer automation in design and construction. The report identifies integration of multi-discipline design and drafting, integration between numerical analysis systems and CAD, and integration for materials management as areas for improving design and construction operations. Other researchers have developed integrated CAD tools for specific construction purposes.

Beliveau et. al. (1993) present a method for using CAD to control material handling on site. Computer graphics and CAD are combined with a dynamic modeler which performs simulation for computer animation. The system uses the dynamic behavior model to visualize material-handling operations using tower cranes.

Cherneff et. al. (1990) present a tool for integrating CAD with construction schedule generation through knowledge-based programming and database techniques. A link is created between the CAD drawing and a knowledge base that is already linked to a project schedule network generator. The resulting CPM networks are reliant on the structure design in the CAD and are directly linked so that any changes in the design result in direct changes to the CPM schedule.

Beliveau (1990) presents a system that uses CAD for automated positioning and control of construction equipment on site. The CAD is integrated to monitor the paths of the equipment, which is tracked by an equipment position locator.

Other research in CAD that is not construction specific but rather process specific can be applied to the integration of CAD for construction purposes. Brown et. al. (1995) present a method for integrating CAD modeling concepts in building design and

CAD linking structures. The resulting system acts as design analysis tool.

Animation using CAD is a very powerful tool that can be used on the construction site. Euler (1994) presents techniques used by RUST International in using 3-dimensional CAD models for advanced animation and visualization on the construction site. The system of CAD integration with site personnel primarily for visualization purposes was found to improve constructability, improve site safety, and reduce construction costs.

Norman (1990) presents AutoMod II, an industrial oriented simulation system that enables users to define the physical elements of a system using CAD-like graphics. The system is composed of two distinct programs. The build portion is used for the physical and logical components of the model. The physical and logical components of the model are compiled into an executable model where the simulation and animation run concurrently. The resulting system is a simulation and animation tool used for manufacturing.

Wichard et. al. (1989) present Construction CAE, a simulation, planning, scheduling, and cost control tool that is integrated with a CAD system. The simulation plan is graphically defined within CAD and is output as real time animation, replayed graphically on the screen. The system has been tested at a nuclear power plant; the animation was found to be a valuable tool for training assistance, studying work details, and addressing safety concerns when planning modifications.

Neural networks are an emerging field of artificial intelligence that mimics the human brain in the way it learns and recalls information. The applicability of neural networks in the construction industry has begun to develop rapidly in the past few years. The purpose of this literature review is to summarize the applicability of neural network use for predicting productivity within the construction industry.

Moselhi, Hegazy, and Fario (1991) present the argument that neural networks offer a better alternative to expert systems for modeling construction operations. They argue that the key difference between the two methodologies is in the way that data is processed. Expert systems are built on expert knowledge criteria while neural networks are models built on actual decisions. Moselhi et. al. state that expert systems address key needs in a decision-making tool, expert knowledge, judgement, and experience. They state that neural networks possess these same benefits as well as the following advantages:

1) a large number of attributes can be considered
2) neural networks learn by example, therefore knowledge acquisition is not difficult
3) data can be classified based on input variables
4) neural network's ability to generalize allows incomplete data sets to be analyzed
5) only a small amount of memory is required as only network weights need to be stored for recall programs

They recommend that neural networks can be used in the construction industry for:

1) selection between alternative construction methods
2) estimation and classification of productivity, cost control, and performance level

4) recognition of construction defects

5) cost escalation

6) optimization tasks

Researchers have presented neural networks for productivity estimation purposes. Neural networks have been developed to predict productivity for pipe handling, pipe welding, concrete formwork placement, consideration of weather effects, trenching, and a desktop excavator.

Knowles (1997) presents neural networks for estimating the productivity of two industrial applications: pipe handling, and piping welding. Knowles presents the factors affecting productivity and the development of a neural network training scheme capable of increasing the abilities of estimators to accurately define a production rate.

Portas (1996) uses neural networks for estimating the productivity of concrete formwork erection. Concrete formwork production has traditionally been a difficult area of construction to predict. The networks were intended to be an aid for estimators. Factors incorporated into the networks include type of formwork, staff, size, location, and site characteristics.

McCabe, Saadi and AbouRizk (1996) use neural networks in order to predict the productivity of trenching and welding activities. The objective of the study is to improve the estimating accuracy of these activities. Factors incorporated for estimating the trenching activity include weather, equipment, shift durations, and the percent of the activity completed. The welding networks incorporated factors of crew size, shift collecting sufficient projects for the piping resulted in poor training of the networks.

Chao and Skibniewski (1994) perform a case study using neural networks to predict the productivity of a desktop excavator. Two neural networks are used for this purpose of the study: One neural network was used to estimate the productivity of the excavator based on conditions such as soil conditions and excavator positioning. The other neural network is used to predict the effects that the interactions between the resources, trucks and excavator, have on the system's productivity. The case study successfully tests the networks and accounts for varying characteristics that affect the productivity of the excavator. Chao and Skibniewski note that additional benefits of neural networks include the fact that qualitative inputs can be quantified, the influence of factors can be better defined, and the combined effects of factors can be accounted for.

Wales (1994) uses neural networks as a means of applying the effects of weather conditions to the labor productivity rates of activities. Daily average temperature, precipitation, and cumulative precipitation over the previous seven days were identified as inputs. The output is a productivity factor that, when multiplied with the average productivity, results in the actual productivity.

2.5 EARTHMOVING OPERATIONS

There has been much work done to improve earthmoving operations, including much work in the simulation field. Shi (1995) presents an automated modeling environment for earthmoving operations; this concentrates on the automated creation of atomic models. It uses a resource based modeling approach. Manavazhi (1997) presents level user descriptions; He uses an earthmoving operation for dam construction as a prototype.

Shi and AbouRizk (1994) study a simulation model for strip mining construction. The work analyzes an earthmoving operation using simulation to determine minimal unit cost of production. It finds that simulation can be applied to equipment intensive operations such as earthmoving at very little cost.

Jawawardane and Price (1994) present an approach for optimizing earthmoving operations which simulates earthmoving using by mass-haul diagrams for design information. The method combines linear programming and simulation. The system input includes cut and fill quantities of different soil types, the locations and soil compositions of potential borrow sites, the available crews, and the required soil composition of the final design. The system output includes recommended construction methods and a sensitivity analysis of variable input parameters.

Hajjar and AbouRizk (1997) present AP2_Earth, a fully functional stand-alone, special-purpose earthmoving simulation system. They define special purpose simulation as a computer-based environment built to enable a practitioner who is knowledgeable in a given construction domain to model a project within that domain in a manner where symbolic representations, navigation schemes within the environment, creation of symbolic model specifications, and reporting are completed in a format native to the domain itself. The approach allows a practitioner to represent the model with elements that are natural to the earthmoving environment. For example, road elements of the earthmoving operation, then simulate it, and output statistics.

CHAPTER 3

SIMULATION MODELS FOR EARTHMOVING

3.1 INTRODUCTION

Computer simulation provides tools that are used to model dynamic interactions between resources within construction activities. Although simulation has gained acceptance in the manufacturing industry, its use in the construction industry is presently limited to academic research. The complexities and time requirements of model development have rendered simulation in the construction industry non-cost effective (Shi and AbouRizk 1994). The Simulation Panel of the Computerized Construction Research Workshop (Ibbs 1987) has identified simplifying existing simulation tools as one of the main issues in the research and development of construction simulation. This chapter presents simulation concepts for earthmoving.

Earthmoving is a common construction operation. Commonly used earthmoving equipment includes bulldozers, excavators, loaders, trucks, scrapers, compactors, graders, and tractors. Five processes can describe the earthmoving process:

1) excavating and loading (referred to as loading)
2) travel-loaded: from the source to the placement
3) dumping
4) placement
5) travel-empty: from the placement to the source

For this work, the placement activity will not be considered. The simple description of the earthmoving process lacks the level of detail required for improving conventional construction planning and estimating techniques. To be accepted into the construction industry, this simplified model must be augmented to improve the current level of detail achieved in the estimating and planning processes. As Figure 3-1 illustrates, the hauling activities can be further expanded to allow for an increased level of detail. The travel-loaded and travel-empty activities have been expanded to incorporate variable site conditions. For example, as construction progresses, the distances of travel in the source and placement areas change. Such a model can accommodate the dynamic nature of an earthmoving construction site.

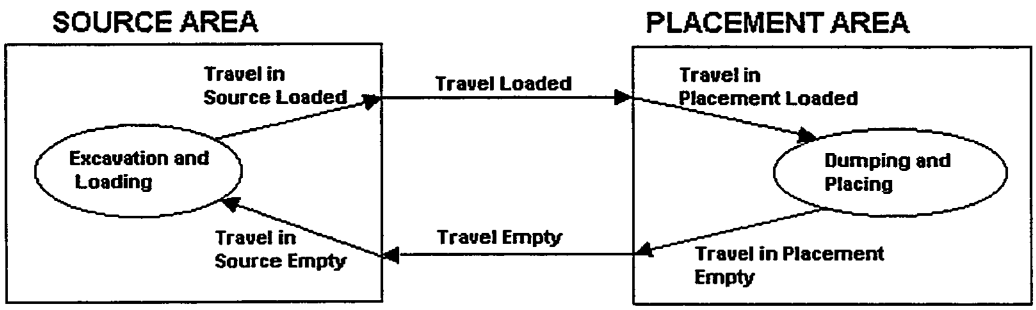


Figure 3-1 Earthmoving Process

There are two main deterministic methods in traditional earthmoving estimating and planning:

1) In the first method, the estimator assumes that unlimited trucks are available and loads the system with enough trucks to ensure that the excavator is constantly busy. Thus the productivity of the system is determined by the productivity of the excavating or loading unit. In this case, the number of trucks is determined by cycle time. The estimator can determine the unit cost of production, because the productivity of the system is known, and the number of resources are also known.

) In the second case, the number of trucks available limits the estimator, the productivity of the system is determined by the cycle time of the trucks and the number of cycles that can be performed in a specific duration of time. Again, the estimator can determine the unit cost of production, because the productivity and the number of resources are known.

Since the first case traditionally provides a lower unit cost, it is in the project manager's best interest to use this type of system. In case (1), the system's productivity is dependent on the productivity of the loading operation. Although much simulation research has focused on the simulation of the entire earthmoving system, little research has been focused on simulating the loading operation. Since the productivity of the system is constrained by the loading operation, this research focuses on the need to simulate these processes. In order to determine the effect of resources interacting at the loading operation, the loading operation must be modeled to the highest level of detail for which data can be collected in a simple and effective manner.

In order to successfully evaluate the construction operation, (1) the overall process must be successfully modeled, to evaluate the required number of trucks and (2) the loading operation must be successfully modeled. Chapter 3 focuses on the simulation of the entire earthmoving process with specialization in the loading operation. In this work, model modularity theory is used to simplify the creation of system models.

[illegible] MODELING THEORY

The accurate modeling of construction operations requires large complex models which are difficult to develop and test. A need to simplify models has been identified as a subject that simulation modeling must address (Ibbs 1987). Graphic user interfaces, model reusability, and modularity have been studied for simplification of the modeling process in construction simulation. Zeigler (1987) and Luna (1992) present hierarchical and modular modeling concepts. One method of modularity, resource based modeling and the use of atomic models in simulation, was introduced by Shi (1995), and found to be a practical form of simulation modeling. In this thesis, atomic models will be used for the description of the earthmoving activities. This allows the creation of complete models using different excavation sub-models, without developing numerous system models.

The atomic model theory is a method of modularity that has been identified as a method of simplifying simulation modeling for construction. The basic components of the modularity concept include the 'atomic model', the 'model base' or 'model library', and 'coupling'. An atomic model can be defined as a basic and unique description of a particular process. The 'model library' consists of a collection of atomic models, which combine to construct a high-level model. Coupling is the act of combining the related models. This chapter presents the creation of atomic models for the special purpose of earthmoving and, in particular, the loading process. Figure 3-2 illustrates the process of coupling atomic models A and B in the construction of a higher level model AB.

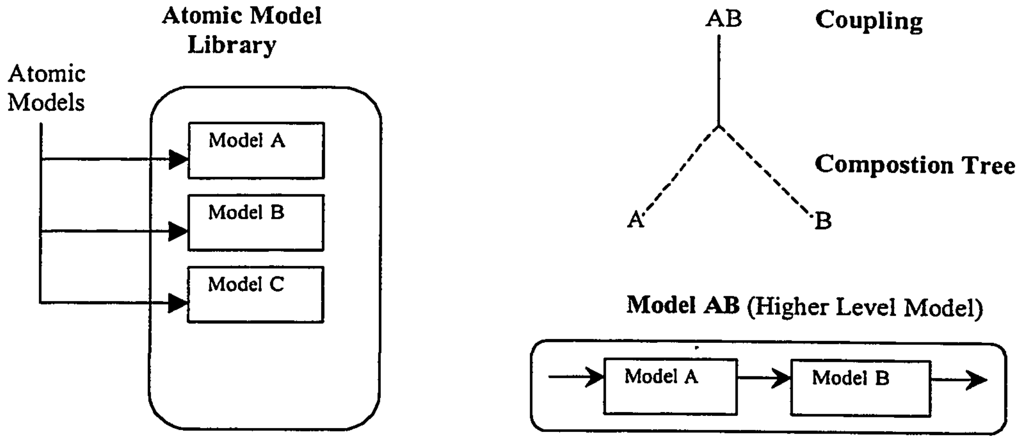


Figure 3-2 Modular Modeling Concepts presented by Zeigler (1987)

3.3 EARTHMOVING ATOMIC MODELS

As described in the introduction, the earthmoving operation is described by four activities. The creation of atomic models for each activity, allows them to be combined using input and output ports to create a system model as described by Figure 3-3.

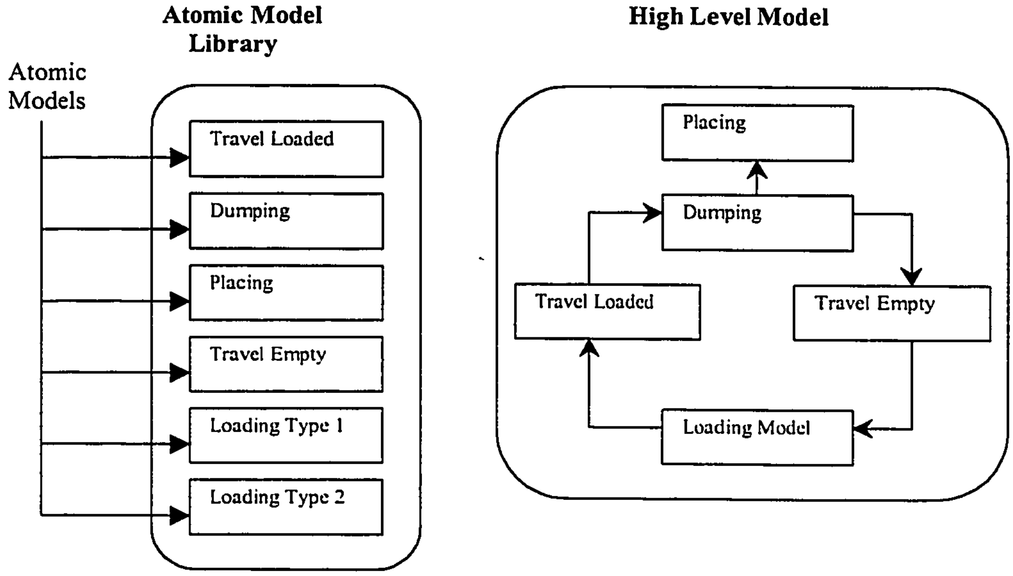


Figure 3-3 Earthmoving Atomic Models

Joining 'atomic' models from the model library creates the high-level system model. The high-level model is composed of the four earthmoving activities, combined in a logical cyclic order. The "Loading Model" in the system model of Figure 3-3 can be considered a variable model for this work. Since there are different loading methods, any of the atomic loading methods from the atomic model library can be placed in the "Loading Model" location.

In the system model, the dynamic nature of the earthmoving operation is modeled by simulation entities which move through the model. The entities used in this model are the hauling units, and for the purpose of this work are trucks. The system model uses discrete event simulation; the dependent variables in the model change at specified points in simulated time.

3.3.1 "AweSim" Models Description

The simulation models used in this work are created in AweSim, an integrated system that uses Visual SLAM Simulation Language. AweSim is a commercial simulation system by Pritsker Corporation. AweSim provides a total system interface that incorporates a network modeler, Visual SLAM, input and output modules, and animation capabilities. For demonstration purposes, this system performs acceptably. In order to accommodate larger models it would be beneficial to bypass the Visual Slam environment. The AweSim system carries excessive overhead features, not required for this work. For all practical purposes, the simulation work contained herein should be integrated into a special purpose stand-alone simulation system such as AP2_Earth.

The AweSim symbols and model components used to develop the models are:

Entities:

Entities flow through a process. They contain attributes that allow a modeler to distinguish between entities in a system. In this case, the entities are the hauling units, trucks that flow through the model of the earthmoving process.

Resource:

A resource is an object, which is required to perform a task. For example, in earthmoving, the loader is a resource. The truck entity requires the loader before it can be loaded. Other resources in the following models include intersections and loading stalls.

XX():

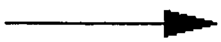
XX() is a global multipurpose variable. For the work herein, the XX() variables provide durations to the activities and are used for controlling the active loading model.

USERF():

The USERF() functions below an activity is a function to assign values to the activity from Visual Basic Code. For the work herein, the USERF() function provides durations to the activities.

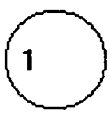
Activity:

The activity is used to delay entities, perform conditional or probabilistic testing, and route entities to non-sequential nodes. An arrow symbolizes an activity.

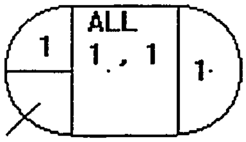


Goon Node:

The goon node passes every entity arriving at it directly through the node.

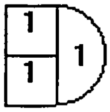


Await Node:



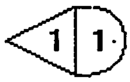
The await node is used to delay entities in a file until the required number of the resource(s) become available. For example, the truck entity may arrive at the loading operation, but if the loader (resource) is busy, then the entity will wait for the loader to be available (free).

Free Node:



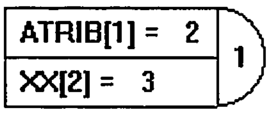
The free node releases a specified number of resource units. An immediate attempt is made to allocate these resources to other entities at AWAIT nodes.

Event Node:



Event nodes call user-written routines with each arrival of an entity. This allows the user to model functions for which a standard node is not provided.

Assign Node:



The assign node is used to change entity attributes or global variables during the modeling.

3.3.2 Travel-Loaded

The travel-loaded atomic model requires knowledge of the road definition. The road as described in Figure 3-4 is broken up into sections. For each section of road, the distance, slope, and road condition may be different. With different properties, the trucks will have different velocities on each section, thus we must simulate the travel on the road on a section-by-section basis.

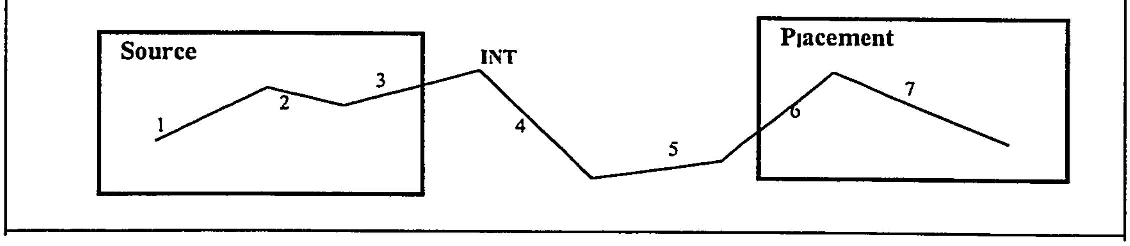


Figure 3-4 Road Description

The descriptive travel-loaded model is displayed in Figure 3-5. Once a truck is loaded, it is sent to the travel-loaded model. In the travel-loaded model, the truck travels on a section by section basis until it has completed the road, at which time it is sent to the dump model. Since there may be intersections along the travel route, they have been incorporated in the model. If there is an intersection at the end of a road section, the truck is sent to the intersection model discussed in Section 3.3.4.

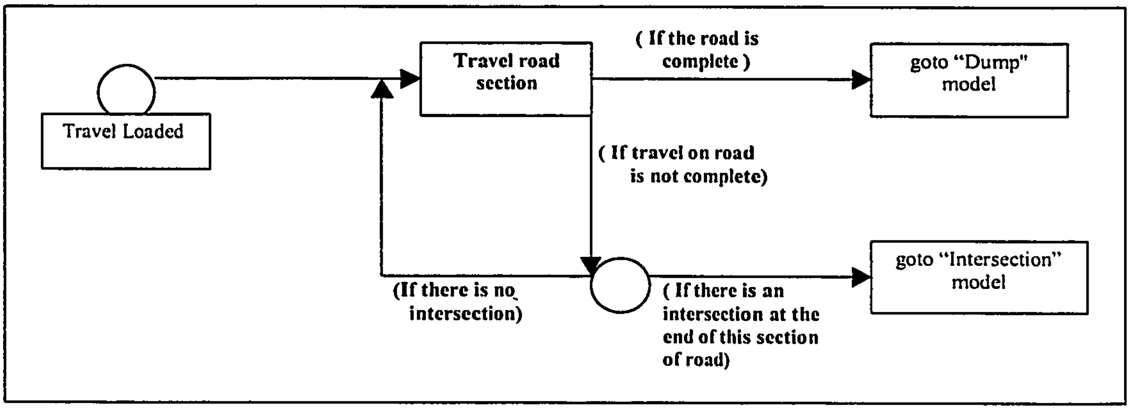


Figure 3-5 Descriptive Model: Travel-loaded Activity

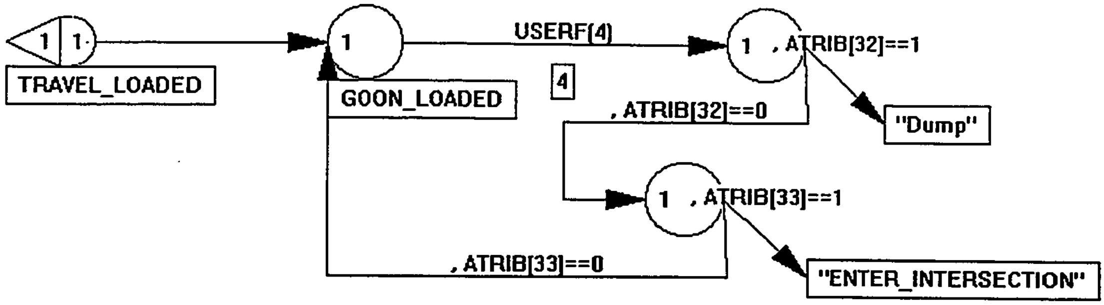


Figure 3-6 AweSim Simulation Model: Hauling-loaded

The AweSim travel-loaded atomic model is shown above in Figure 3-6. The activity USERF(4) is a user written function that determines the time that it takes the truck to travel that section of road. If the truck has completed the road, then ATRIB(32) is set to one and the truck is sent to the dumping model. If the road has not been completed, ATRIB(32) is set to zero and the truck is scheduled to either cross an intersection or travel the next section of road. If there is an intersection at the end of the current section of the road, then ATRIB(33) is set to one, otherwise it is set to zero. Once the truck has crossed the intersection, it is returned to the travel-loaded model to complete the road.

3.3.3 Travel-Empty

The travel-empty model functions in almost the same manner as the travel-loaded model. However, the trucks travel in the opposite direction. The other difference is that the trucks are placed in the travel-empty model from the dump model, and at the end of the road, the trucks are sent to the loading model.

3.3.4 Intersection

When a truck entity enters the intersection model, the truck wants to cross the intersection as soon as possible. The intersection is characterized as a resource. This means that when the truck arrives it has to wait for the intersection resource to be free.

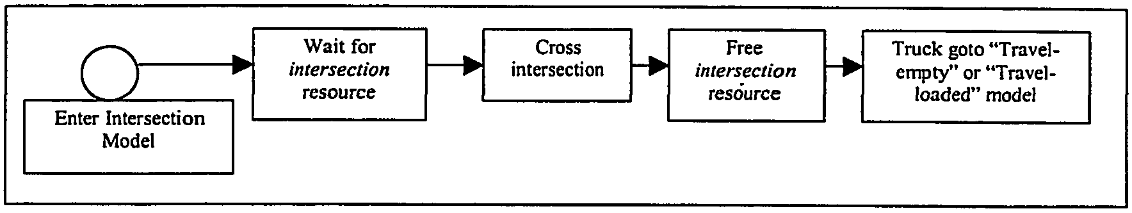


Figure 3-7 Descriptive Model: Intersection

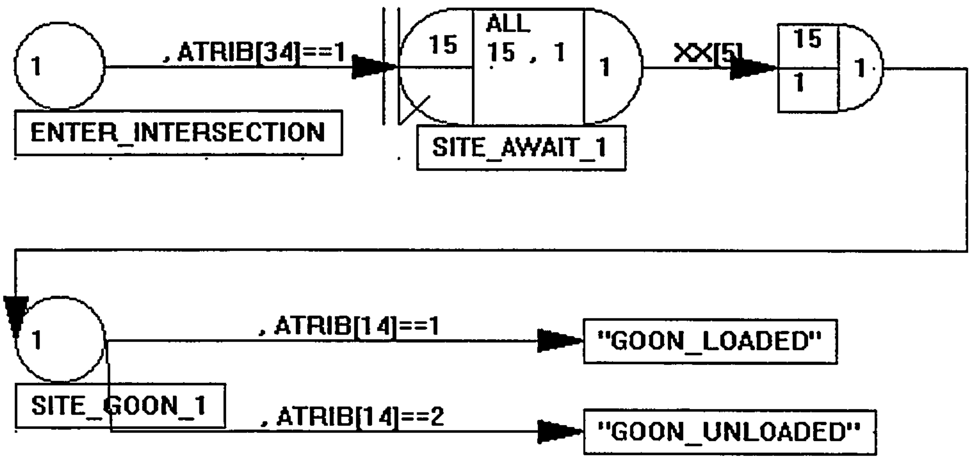


Figure 3-8 AweSim Simulation Model: Intersection

Once the intersection is free, the truck will utilize the resource and cross the intersection, an activity that takes a duration XX(5). Upon crossing the intersection, the intersection resource is freed for other entities to use, and the truck is sent back to the travel model truck is loaded or empty, the intersection model knows which model to send it to.

External traffic can be incorporated into the intersection through a create node which can add traffic at random intervals. When external traffic enters the intersection, it utilizes the resource in the same manner as a system entity (truck). The external traffic AweSim model is displayed in Figure 3-9.

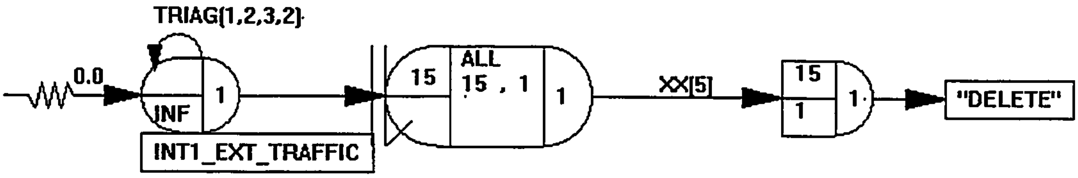


Figure 3-9 AweSim Simulation Model: Intersection External Traffic

3.3.5 Dumping

For the purpose of this thesis, a detailed model of the dumping process is not required. The dumping model is represented by a single activity represented with a random distribution.

3.3.6 Loading Direction

Once the trucks arrive at a loading location, they need to be directed to the proper loading model. The trucks are sent to the appropriate loading model using a feed forward directive approach. Each available loading model is labeled, and the loading direction model sends the truck entities to the active loading model; this is described by the variable XX(1) as shown in Figure 3-10.

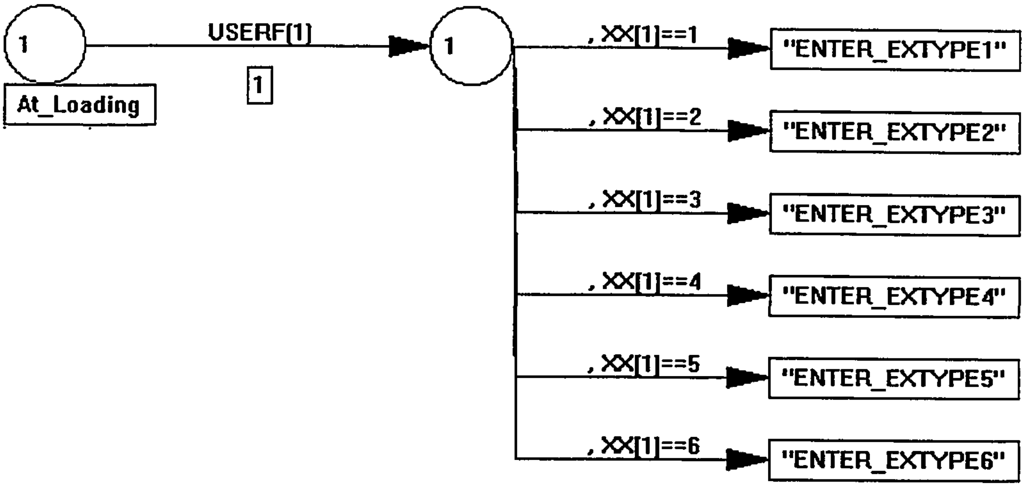


Figure 3-10 AweSim Simulation Model: Directive Loading Routing

3.4 EXCAVATION ATOMIC MODELS

The modeling of the excavation and loading process is extremely important in terms of the system's productivity, because the interactions with the excavator affect the maximum productivity of the system. Often in earthmoving, the maximum (or near maximum) productivity of the system results in the lowest unit costs. Since the maximum productivity of the system is constrained by the loading unit, one can assume that the interactions between resources at the loading location play an important role in determining the system productivity.

The work focuses on the excavation process using hydraulic excavators. Two types of excavators are studied, front loading shovels, and track model backhoes.

The following work describes the different situations that can arise at the loading location. Three loading situations have been identified for shovels, and three loading situations have been identified for backhoes. The models are developed on the premise easy data collection from site. For example, the loading of a truck activity is not described by simulating the individual bucket loads of the excavator. This method was tested, but data collection turned out to be ineffective on a large-scale basis.

The loading activity is defined by the time it takes the excavator to load the truck once the truck is in position. In the AweSim simulation models, the loading activity is represented by the USERF(2) function. Since this loading activity will vary depending on the relative positioning of the trucks and loading units, and the soil types, another approach is used to determine this duration. Neural networks are used to determine the loading time based on the characteristics of the site conditions, relative loader/truck positioning and the soil types. For large-scale data collection, the neural network approach was found to be more effective than collecting data for simulating motions of the excavator (Chapter 5).

Since the system productivity is primarily based on the excavator unit, it is desirable to simulate this action as accurately as possible. The six different loading methods have been identified, include three for shovels, and three for backhoes.

1) Shovel and trucks: Loading on both sides
2) Shovel and trucks: Loading on one side
3) Shovel and trucks: Drive-by loading
4) Backhoe and trucks: Top-loading, back in
5) Backhoe and trucks: Top-loading, drive-by
6) Backhoe and trucks: Level-loading

... through an informal survey of an earthmoving contractor's project management personnel and verified through on site monitoring. When estimating and planning, these types of loading situations are typically considered before allocating an average productivity to the loading resource.

Values for the durations of the activities within the loading models have not been described but should be collected for the different excavators, and loading scenarios.

3.4.1 Shovel and trucks: Loading both sides

The first, most effective, and most common loading operation using a shovel type excavator is the loading of trucks on both sides of the excavator, as illustrated in Figure 3-11. When a shovel is loading with trucks on both sides, four activities characterize the process:

1) The truck arrives at the loading location and positions to back into a loading stall.
2) The truck waits for a loading stall resource to become free. Once a loading stall becomes free, the truck backs in to the loading stall.
3) Once in the loading stall, the truck waits for the shovel resource. Once the shovel resource has been allocated to the truck, the truck is loaded. Upon completion of the loading, the shovel resource is freed.
4) After the truck has been loaded, it exits the loading stall. Once it has exited the loading stall, it frees the loading stall resource.

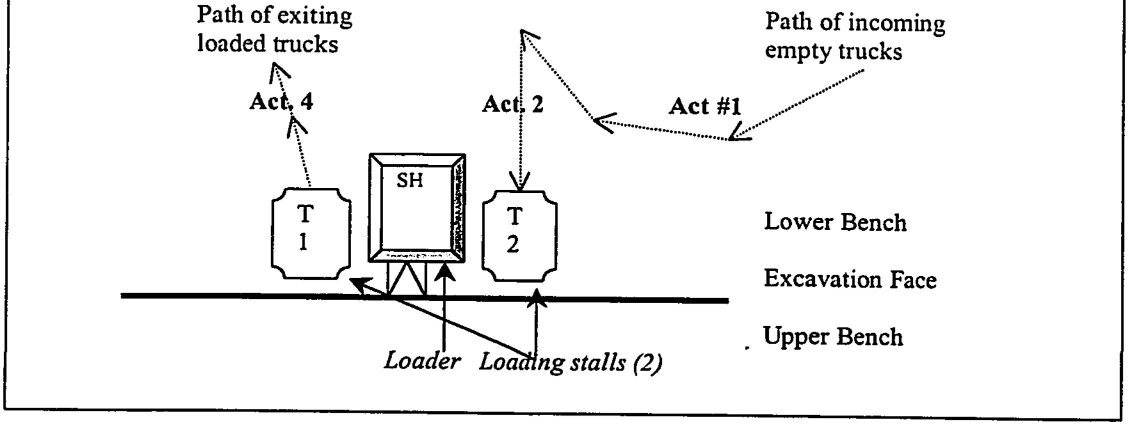


Figure 3-11 Loading Layout: Shovel and Trucks Loading Both Sides

Figure 3-12 visually describes the process. Figure 3-13 illustrates the AweSim simulation model of the loading process.

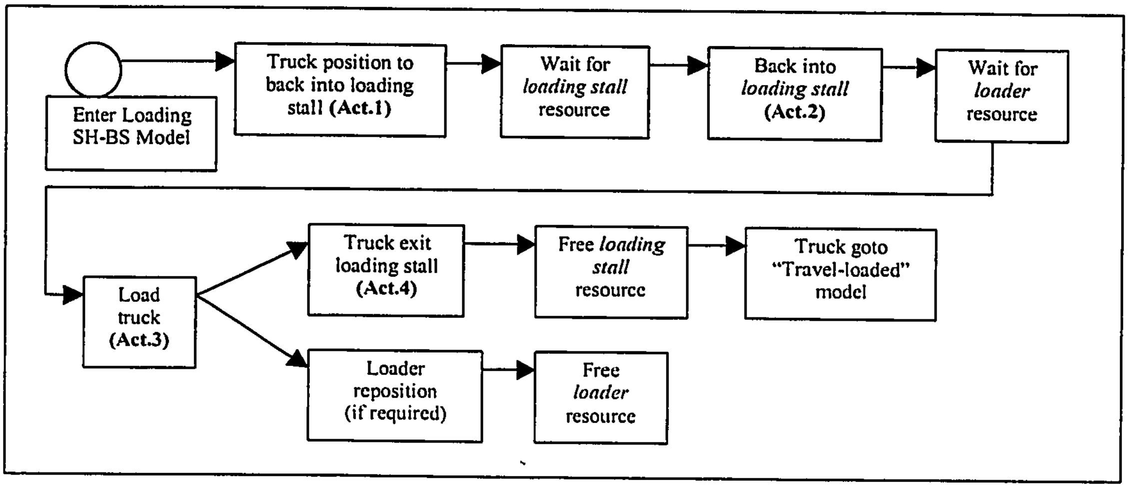


Figure 3-12 Descriptive Model: Shovel and Trucks Loading Both Sides

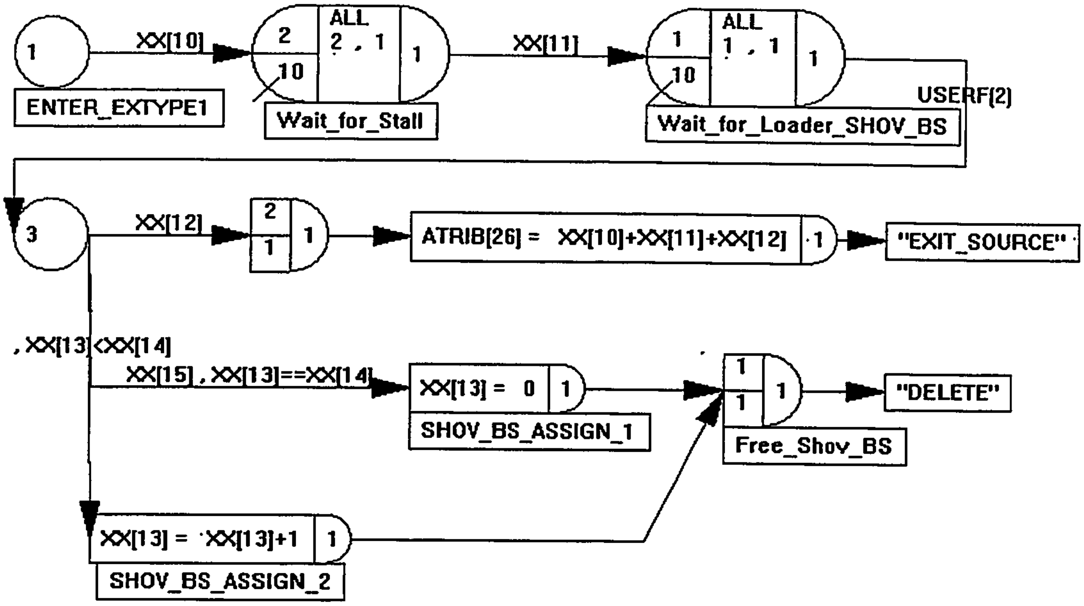


Figure 3-13 AweSim Simulation Model: Shovel and Trucks Loading Both Sides

3.4.2 Shovel and trucks: One-side loading

The shovel and trucks loading one-side scenario results in the same simulation model as Figure 3-13, but there is only one loading stall resource. The result is that after a truck is loaded, the loader will have to wait for the loaded truck to exit the loading stall and the empty truck to enter into the loading stall. This loading situation is much less efficient as the shovel and trucks loading two stalls, described in section 3.4.1.

3.4.3 Shovel and trucks: Drive-by loading

The truck and shovel drive-by operation involves trucks parking in behind the shovel, as illustrated in Figure 3-14. In drive-by loading the shovel faces the excavation face, and the trucks pull up parallel to the wall behind the shovel for loading. The advantage to this loading situation is that there is little positioning time for the trucks as they line up for the loader. The disadvantage is that productivity decreases, because the loader must pivot 180 degrees from the excavation face in order to load the trucks, thus the loading activity duration is longer. Figure 3-16 illustrates the AweSim simulation model for the drive-by loading operation. The USERF(2) function calculates the loading activity duration.

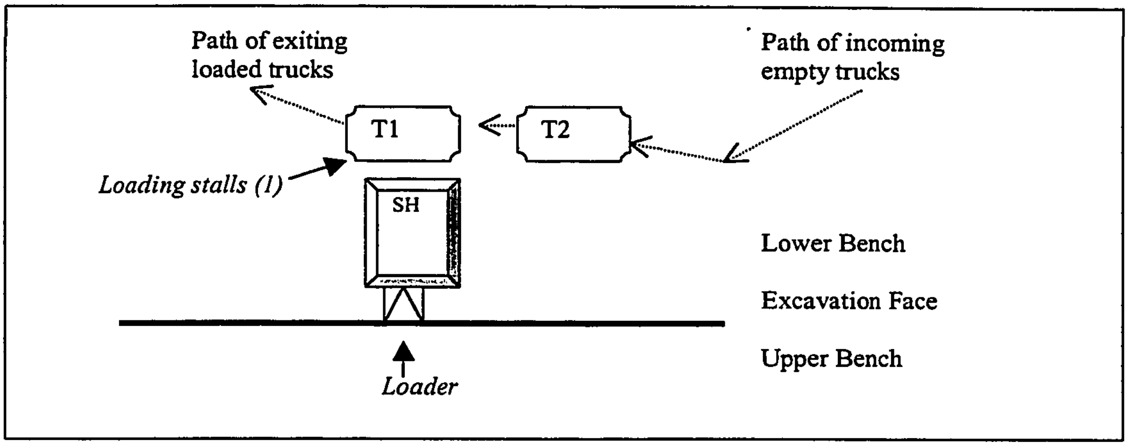


Figure 3-14 Loading Layout: Shovel and Trucks, Drive-By Loading

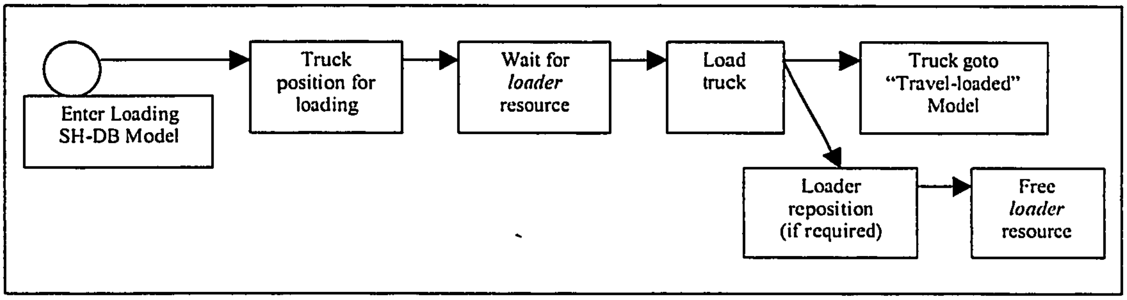


Figure 3-15 Descriptive Model: Shovel and Trucks, Drive-By Loading

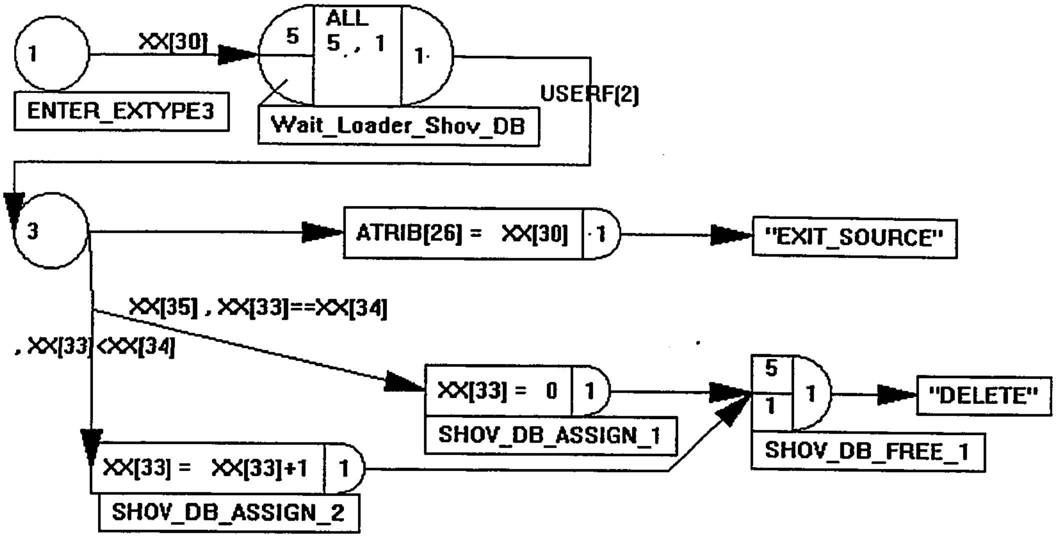


Figure 3-16 AweSim Simulation Model: Shovel and Trucks, Drive-By Loading

3.4.4 Backhoe and trucks: Top-loading

In the backhoe and trucks top-loading operation the backhoe sits on the bench above the trucks, as illustrated by Figure 3-17. The trucks back in to the loading stall below the backhoe, where they are loaded. This method is beneficial because the backhoe operator can clearly view the interior of the boxes of the trucks, ensuring that they are loaded fully. Figure 3-18 visually describes the simulation model.

The backhoe and trucks, top-loading operation is described using 4 activities:

1) The truck arrives at the loading location and positions itself to back into a loading stall.
2) The truck waits for the single loading stall resource to become free. Once the loading stall becomes free, the truck backs into the loading stall.

3) Once in the loading stall, the truck waits for the backhoe resource. Once the shovel resource has been allocated to the truck, the truck is loaded. Upon completion of the loading, the backhoe resource is freed.

4) After the truck has been loaded, it exits the loading stall. Once it has exited the loading stall, the loading stall resource is freed.

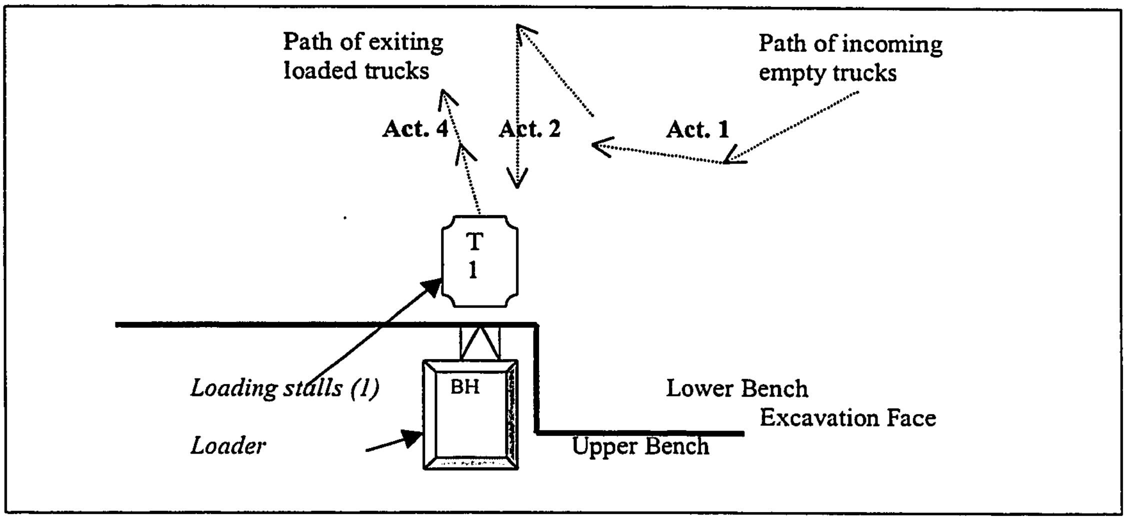


Figure 3-17 Loading Layout: Backhoe and Trucks, Top-loading

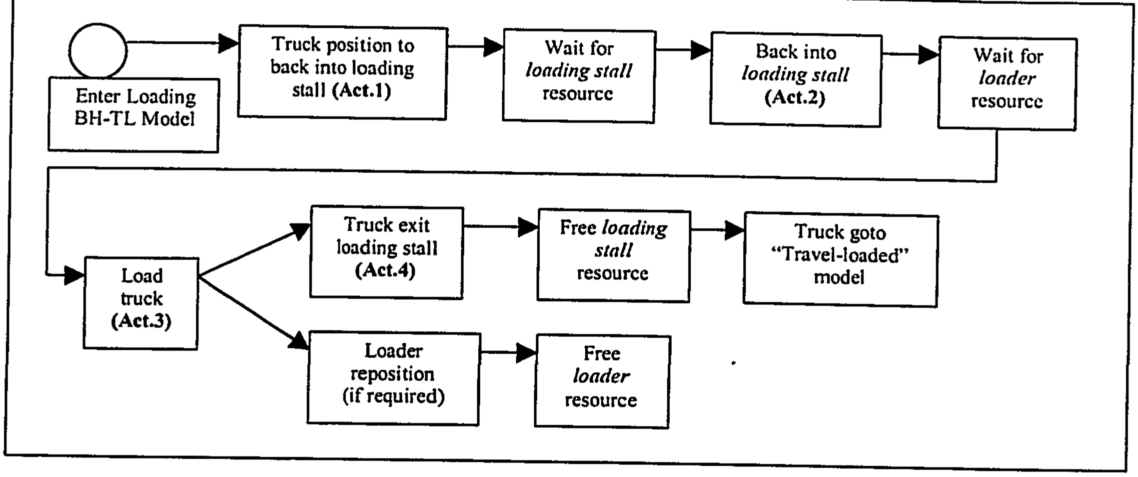


Figure 3-18 Descriptive Model: Backhoe and Trucks, Top-loading

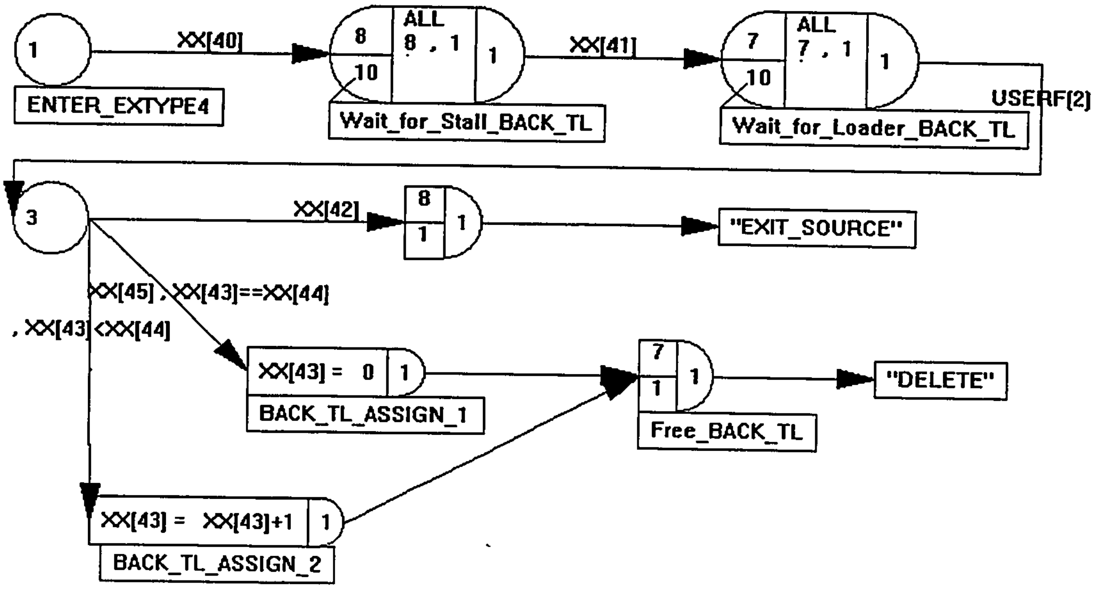


Figure 3-19 AweSim Simulation Model: Backhoe and Trucks, Top-loading

3.4.5 Backhoe and trucks: Top-loading, Drive-by

In the backhoe and trucks top-loading, drive-by operation, the backhoe sits on the bench above the trucks. The trucks drive forward into the loading stall and position themselves directly below the backhoe as illustrated in Figure 3-20. This method presents many benefits. The backhoe operator can clearly view the interior of the boxes of the trucks, ensuring that they are fully loaded. The trucks also have no pre-loading activities. This loading situation requires a straight cut face, which is not always possible, particularly in pits. The other advantage is that the backhoe does not have to wait for the trucks to position themselves, as they are lined up behind each other. This is considered by earthmoving construction managers to be the most effective method of loading when using a backhoe. A descriptive model of the operation is displayed in Figure 3-21. The AweSim simulation model is displayed in Figure 3-22.

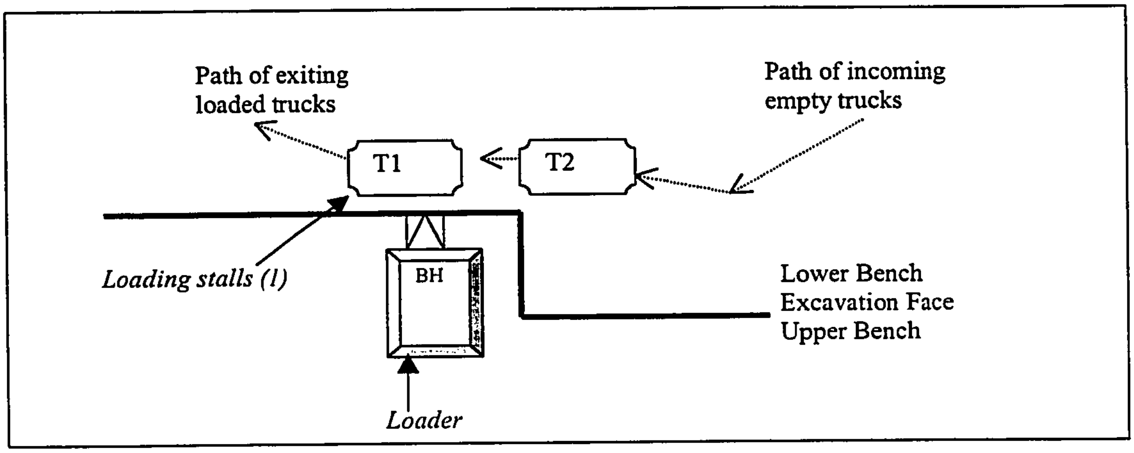


Figure 3-20 Loading Layout: Backhoe and Trucks, Drive-By Loading

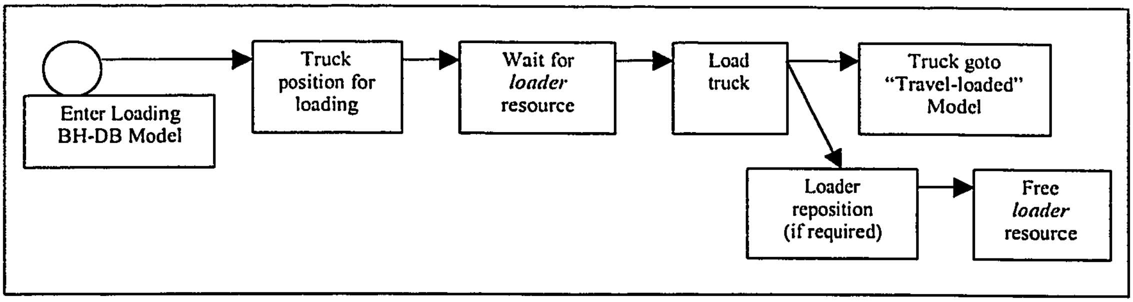


Figure 3-21 Descriptive Model: Backhoe and Trucks, Drive-By Loading

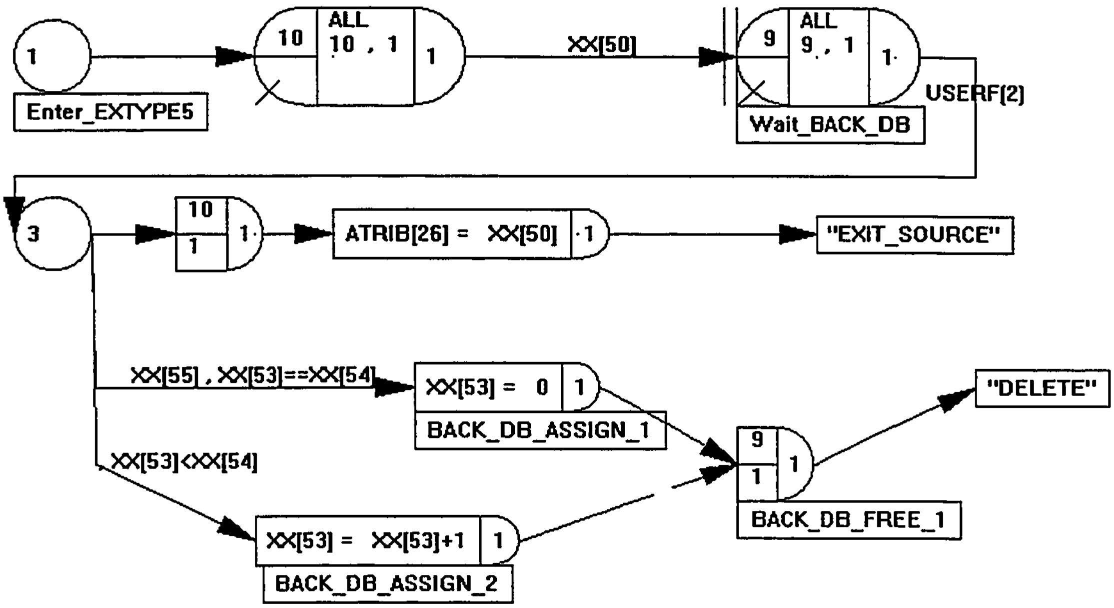


Figure 3-22 AweSim Simulation Model: Backhoe and Trucks, Drive-By Loading

3.4.6 Backhoe and trucks: Level-loading

The backhoe and trucks level-loading operation is characterized by the backhoe and trucks being located on the same elevation, as illustrated in Figure 3-23. This situation is very inefficient. The trucks must back up to the backhoe so that it can place loads in the open end at the rear of the truck. Also, the swing angle of the backhoe is approximately 180°; this reduces the excavator's productivity. As well, the boom of the backhoe must extend upwards to load the truck. The loading operation involves four steps:

1) The truck arrives at the loading location and positions itself to back into a loading stall.
2) The truck waits for the single loading stall resource to become free. Once the loading stall becomes free, the truck backs into the loading stall.

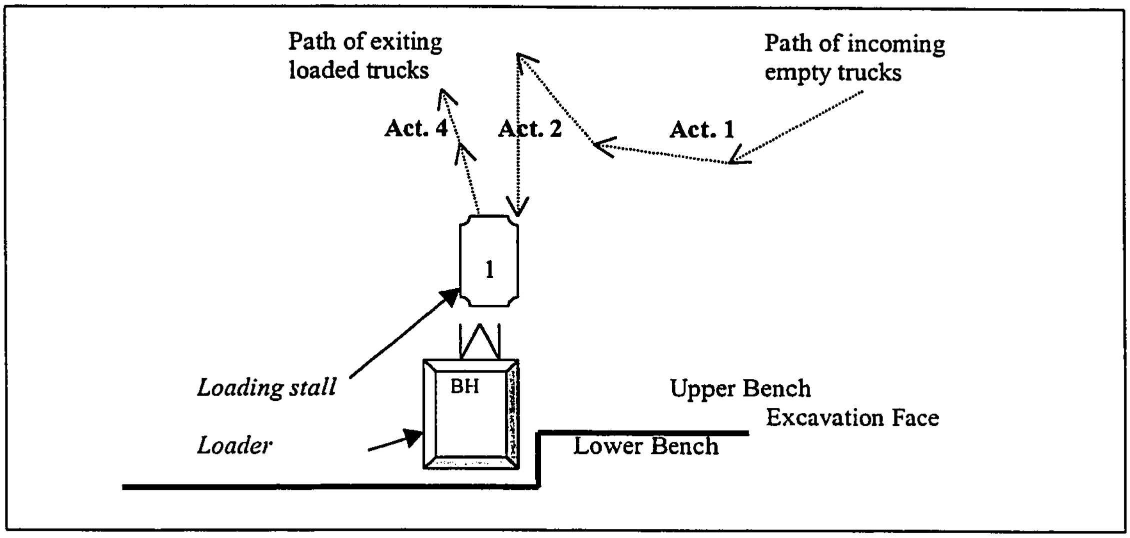


Figure 3-23 Loading Layout: Backhoe and Trucks, Level-loading

3) Once in the loading stall, the truck waits for the backhoe resource. Once the shovel resource has been allocated to the truck, the truck is loaded. Upon completion of the loading, the backhoe resource is freed.
4) After the truck has been loaded, it exits the loading stall. Once it has exited the loading stall, it frees the loading stall resource.

Figure 3-24 descriptively illustrates the simulation model, Figure 3-25 displays the AweSim simulation model for this operation.

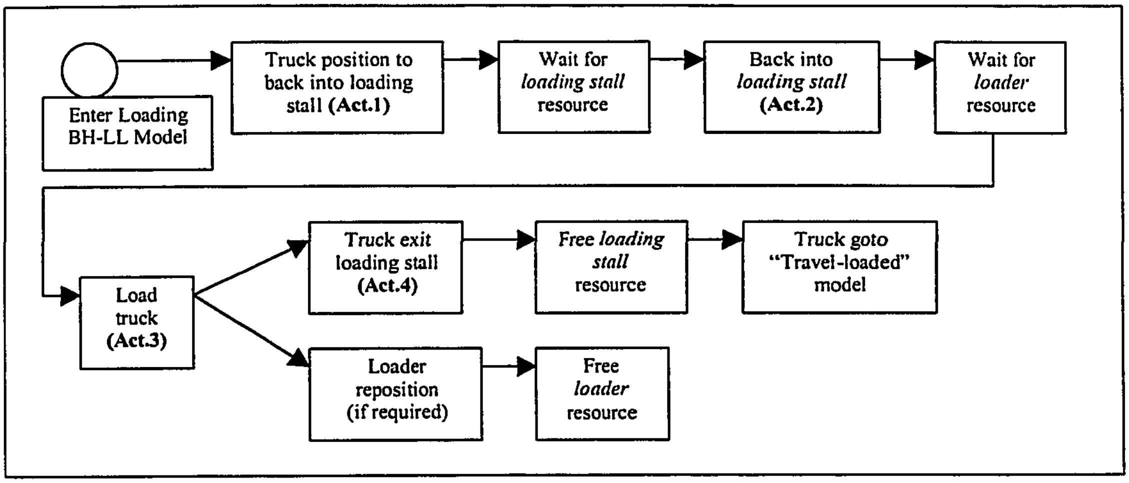


Figure 3-24 Descriptive Model: Backhoe and Trucks, Level-loading

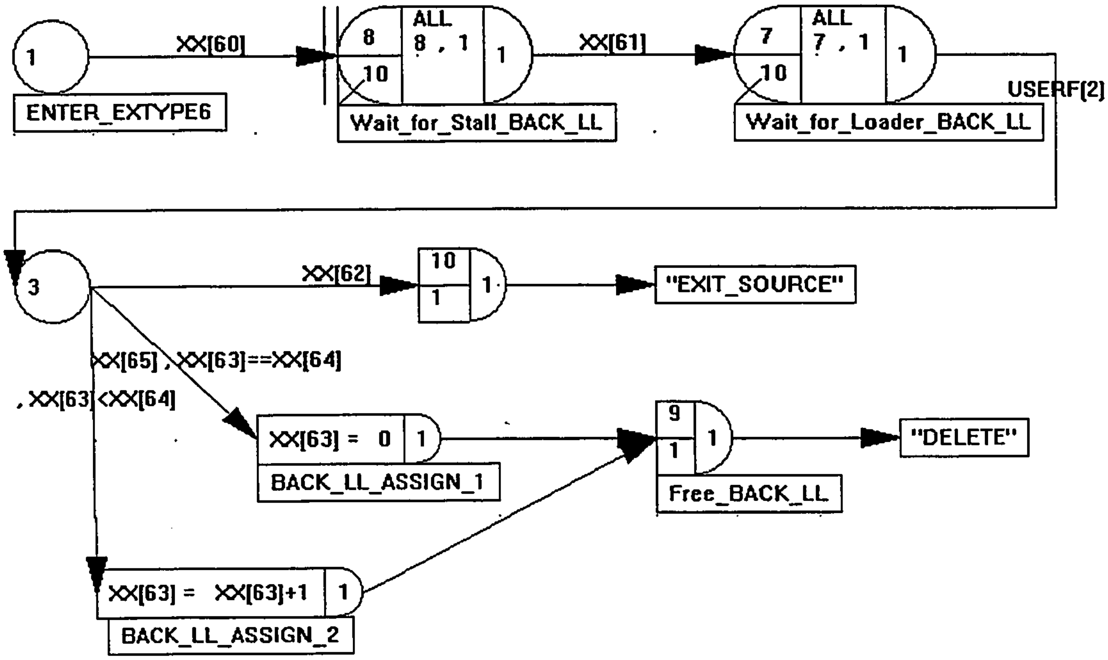


Figure 3-25 AweSim Simulation Model: Backhoe and Trucks, Level-loading

3.3 ATOMIC MODEL LINKING

The atomic models are combined using coupling to create a high-level (or system) model. Coupling is the act of combining the atomic models. Coupling is performed using linking structures and communication ports. Atomic models are combined through input and output ports and the linking structures, represented by arrows, as illustrated in Figure 3-26.

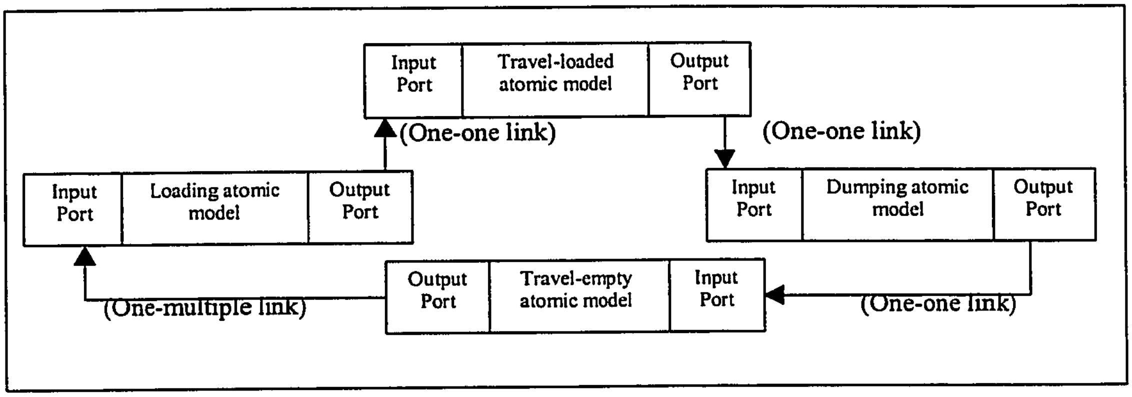


Figure 3-26 High-Level Earthmoving Simulation Model

Communication ports identify how atomic models interface with one another. Input ports receive entities from other models. Output ports send entities to other models, where the entity is received by an input port. The earthmoving models uses labeled input and output communication ports. That is the input node of the atomic model receiving the entity is described with a label, and the output port sends the entity to the input port by directing the entity to this label.

Shi (1995) presents that linking structures between atomic models can be divided into direct and indirect linking structures. Direct structures do not alter simulation entities during the linking operation. Indirect linking structures are used to couple models where entities are not compatible. For example, if the entity leaving an atomic model A is measured in units X, and the simulation entity in model B is measured in units Y then entities can not directly flow from one atomic model to the next. For the purpose of this work, the trucks are the entities throughout, therefore, direct linking structures are used for linking the atomic models.

Two types of direct linking structures are used for the earthmoving process described in this Chapter. One-one links are used to combine models where one model is the sole input to another model. For example, a one-one link is used to link the travel-loaded model to the dumping model. A one-multiple link is used when multiple models follow a single model, and only one of them can be released at a time. For example, the one-multiple link is used to link the travel-empty model to the loading model. The one-multiple link used in the earthmoving model is the loading direction model described in Section 3.3.6. From the loading direction link, the entity is sent to only one of the loading models.

The advantage of using atomic models for the high-level model is that atomic models are independent from each other. The models can be defined in any order, using the linking structures, to produce a high-level model. This allows the models to dynamically change throughout the simulation run. For example, the loading model in Figure 3-26 is variable. Any of the six different loading models described in this Chapter can be placed in the loading model, thus creating a dynamic system model.

Figure 3-27 illustrates a system model developed from the atomic models. The loading model displayed is the backhoe and trucks drive-by loading operation. The model displayed is a creation of the atomic models that have been combined using

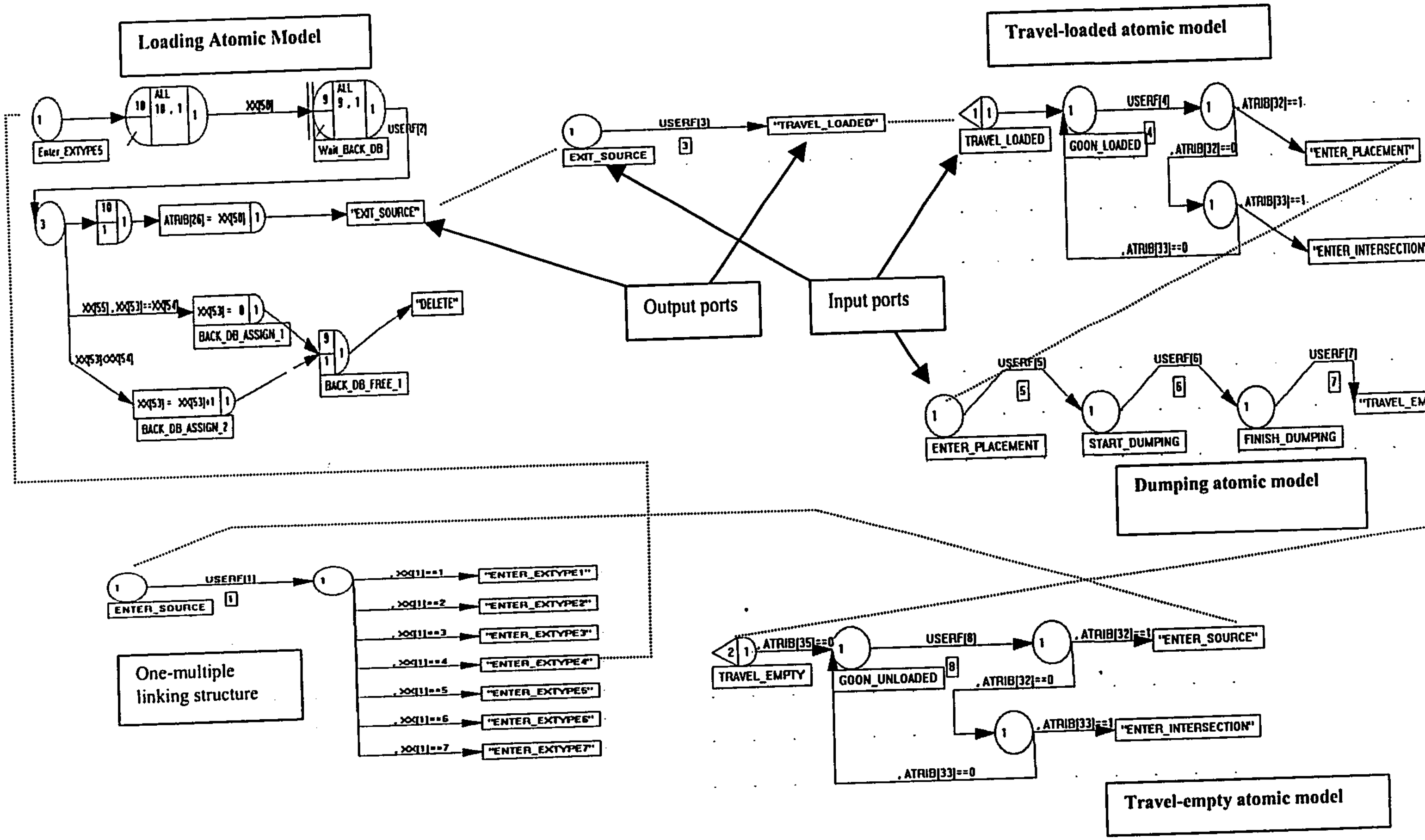


Figure 3-27 Earthmoving System Model

linking and communication ports. In Figure 3-27, the linking structures are illustrated by the dotted lines.

3.6 CONCLUSIONS

Traditional methods of earthmoving estimation and planning are deterministic, and fail to consider the dynamic and random nature of construction operations. The simulation system allows for the analysis of earthmoving to consider the randomness of activities and the interactions between equipment.

This chapter provided a method of using modular concepts to develop an earthmoving system model. Atomic models combine to create the system model. This allows the system to use different atomic loading models, but does not require the development of a large, complex model for each type of loading operation. The modular concept was successfully used to simplify the modeling of the earthmoving process.

Modeling the earthmoving loading operations at detailed levels allows the effect of interactions between the trucks and the loading unit to be analyzed and quantified. The effects that the interactions have on the system productivity can be observed. Using different loading operations, when simulating an earthmoving system allows the different loading types to be compared.

CHAPTER 4

CAD-BASED SIMULATION MODELING

4.1 INTRODUCTION

Computer Aided Design (CAD) systems are used by engineers and draftspersons as tools for design and drawing. CAD systems are now available on personal computers, making them cost effective for use on construction projects of all sizes. These systems are currently used primarily for design. Two of the more popular CAD platforms are MicroStation and AutoCAD. This thesis uses MicroStation for the CAD platform, although the theory still applies to AutoCAD. These CAD programs are used extensively for pre-construction design, but are seldom being used during the construction phase.

Choi and Ibbs (1990) identified the need for quantity data extraction from CAD for integration in project-wide databases. Manavazhi (1997) and Shi (1995) identified the need to integrate simulation with CAD to define the site and extract design configurations and information. It is necessary to integrate CAD and simulation models because an effective simulation model requires the addition of a dimension to the estimating or control process that traditional analysis methods do not provide. In this case, the goal is to increase the accuracy of the simulation model results. CAD can tremendously increase the amount of information provided to the model. This information can then be used to more accurately simulate the construction operation. This Chapter focuses on a CAD-based simulation modeling methodology to address the needs identified by these researchers.

The integration of CAD and simulation models for construction is an unproven topic that presents two major benefits: The first benefit is the ability to simplify the simulation modeling environment for the user. The user can use the CAD environment to define the high-level simulation model from predefined atomic models, thus minimizing the user's required simulation knowledge. The second benefit is that CAD-based modeling can incorporate site specifics and immense amounts of data. This adds the ability to evaluate the system on a more detailed level than existing methods of analysis. Much research into the integration of CAD and simulation has focused on visual presentation and animation, but has not focused on the interactions and processes involved in construction.

The vast design information stored within CAD files creates logic that by integrating simulation with CAD can provide time savings in defining the project and site for the simulation. This Chapter presents a methodology for simulation modeling integration with CAD concentrating on the CAD portion of the modeling. Chapter 3 presents the simulation corresponding to the CAD-based simulation system for earthmoving operations, used throughout this chapter for an example. Chapter 6 presents a sample application of this theory for a special purpose simulation system.

4.2 BASIC CAD TERMINOLOGY FOR MICROSTATION

MicroStation is a very in-depth program that requires extensive knowledge to use efficiently. This section lists and explains some of the basic terminology and general configuration of MicroStation to give the reader some basic background.

4.2.1 Basic Terms

File Types: CAD design files can be either 2 dimensional (2D) or 3 dimensional (3D). The file type is defined during the creation of the CAD file. 2D CAD files contain the entire design on one drawing plane. 3D CAD files allow the design drawing to exist as a 3 dimensional structure in much the same way as it would exist in real life.

High Level File Description: A CAD design file consists of an internal hierarchy as illustrated in Figure 4-1. Each file is composed of drawing levels. Elements (or drawing objects) are placed on a specific level and exist solely on that level. Each element contains individual attributes and properties.

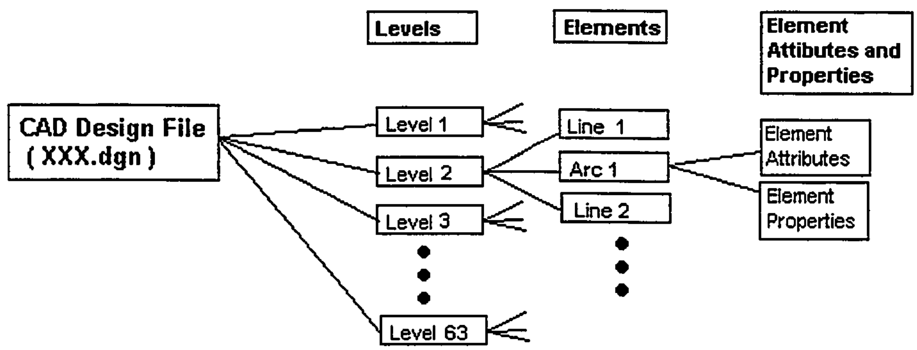


Figure 4-1 MicroStation File Breakdown

Level Structure: Each CAD file is composed of drawing levels. MicroStation contains a set number of levels (63). Each level is independent from the others and is in itself either 2D or 3D corresponding to the file type. Any combination or assortment of levels can be viewed simultaneously. This allows the user to view drawing details in congested areas. When a drawing element or object is placed in MicroStation, it is placed on a single level. Each element placed on a specific level maintains its own attributes. The description of drawing elements can be found in Section 4.2.2. The Level Display Control in Figure 4.2 currently displays the highlighted levels, 34, 35, 36 and 59. The active level, 59, is indicated by the circular highlight. The active level is the level on which elements are placed. By maintaining standard practices throughout designs, the levels in which specific aspects of drawings are placed can be kept consistent. For example, in earthmoving, level 1 can be reserved for original ground.points, level 2 for the original ground developed surface, and so on. In the displayed example, level 34 is used for the excavation limits, level 35 for the excavation side slopes, and level 36 for the excavation bottom outline. Although all this information can be stored on one level, if it is stored on three levels, it is easier for editing and maintenance.

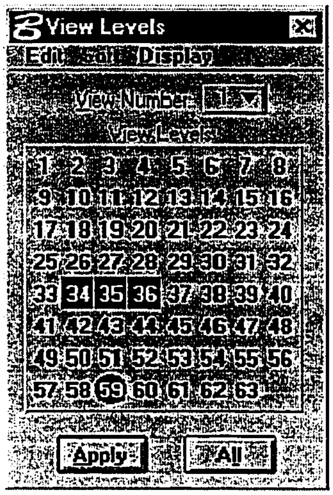


Figure 4-2 MicroStation Level Display Control

Point: A point is a zero length line, possessing X, Y, and Z coordinates. It is used for element entry, referencing, and element selection. Pressing the right mouse button in CAD defines a point.

Tentative Point: A tentative point is a form of graphic input that is used to preview the location of the next data point, or is used to define a point of reference for entry of the next data point (Bentley, 1995).

Reset: A reset is an action that allows the user to release control of the active element or tool. This action is used to prevent incorrect actions and selections.

4.2.2. Element Environment

Drawings in CAD are created using elements. An element is the smallest definable drawing unit available in MicroStation. Each element possesses its own individual attributes and properties. Drawing objects are created using either a single element or a combination of elements, with each element maintaining its own properties and attributes. Drawing objects are used by MicroStation to simplfy the drawing process.

Element Types: The different types of elements are CellLibraryHdr, CellHeader, Line, LineString, Shape, TextNode, Curve, ComplexString, Conic, ComplexShape, Ellipse, Arc, Text, Surface, Solid, BsplinePole, PointString, Cone, BsplineSurface, BsplineBoundary, BsplineKnot, BsplineCurve, BsplineWeight, Dimension, SharedCellDefinition, SharedCell, MultiLine, Tag, RasterHeader, and RasterComponent. Although the various element types seem confusing, knowledge of them is not important for drawing. Pre-defined drawing object tools discussed later are used for drawing. Element types do become important for data extraction methods described later. Special purpose elements can be created, but the existing available elements should suffice for most practical applications.

attributes include the level, color, line weight, line style, file type and color (for closed elements), and class (primary or construction). Attributes of an element are determined by the active settings. For example, while the active color is set to red, the color attribute on a newly placed element is red. Changing an active setting has no effect on previously placed elements, however, tools are available for changing attributes of previously placed elements.

Element Properties: Each element possesses a property referred to as the origin. The origin of an element is defined by the X, Y, and Z coordinates. Different elements have different origins with respect to the location and shape of the element. The element origin is an important property, since external interfacing with elements is often done through the element origin.

Tools: Since the creation of objects and drawings in CAD is designed to be quick and easy, there are many predefined drawing tools. Specific tools and toolboxes that place and combine elements in easy to use formats are used. Tools either place or combine elements as drawing objects in quick and easy drawing formats. Tools are represented in toolboxes by icons. For simplicity, the term "tool" is used to refer to a tool and its icon. Figure 4-3 illustrates the toolboxes "Linear Elements" and "3D View Control." Toolboxes are available for drawing, view control, editing, dimensioning, and inserting text, cells, and tags.

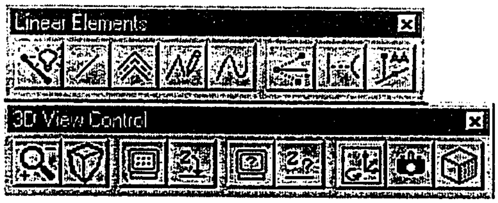


Figure 4-3 MicroStation Drawing and View Control Toolboxes

4.3 CAD-BASED SIMULATION MODELING METHODOLOGY

4.3.1 Introduction

The need to take into consideration detailed aspects of both the site and structure of the project introduces the complexity of defining this information for the simulation. CAD integration has not been used to address this need in any earlier work in construction simulation.

The CAD environment was designed for engineering and drafting purposes and thus does not necessarily lend itself as an ideal environment for simulation definition. Since CAD users often require special purpose utilities, it was designed with the approach that "add-on" programs should be easily developable. To enable the creation of add-on programs, three types of interfaces are available to users. Using CAD specific interfaces, the CAD platform becomes a tool for defining the site specifics for the simulation environment.

By integrating CAD with simulation, the task of data input and site description for the simulation model can be semi-automated. With a drawing of the design available, it also makes sense that the construction sequencing can be defined from within the CAD environment. By incorporating the atomic models and high level models, the be extractable to simulate the construction.

Integrating CAD and simulation develops the CAD-based simulation modeling methodology and is illustrated in Figure 4-4. The simulation module is separated from the CAD module to maintain individuality and limit the simulation knowledge required by the user. The simulation and CAD modules are connected by an intelligent data manager (Appendix D). The simulation model is created using modular simulation models. A sample set for earthmoving is described in Chapter 3.

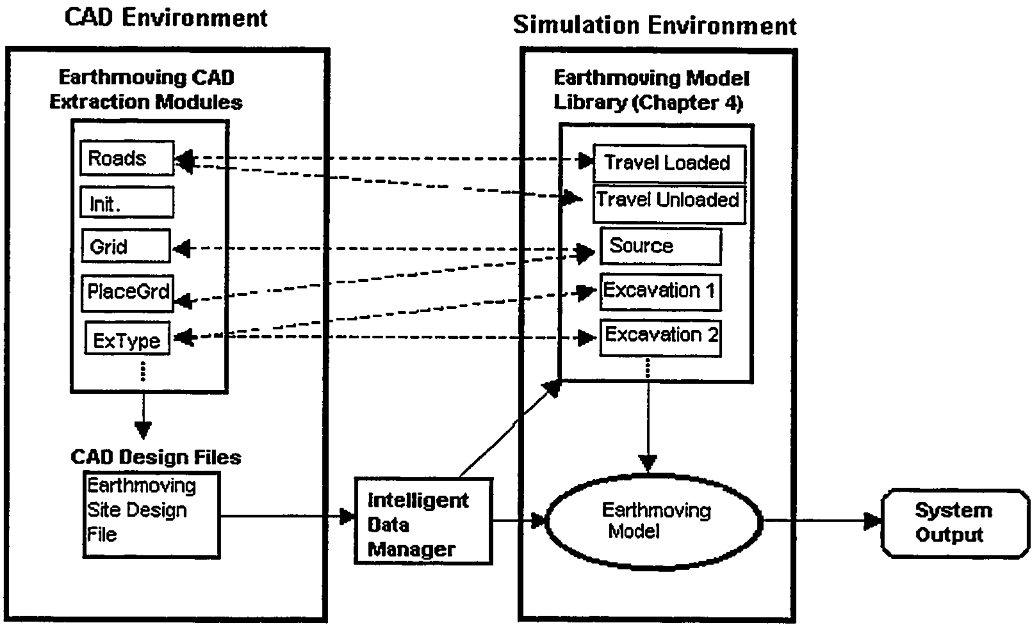


Figure 4-4 CAD-Based Simulation Methodology

CAD Environment: The CAD module contains the design file(s). The design file(s) should contain all relevant information for the regarding the site and the design of the structure to be constructed. The "CAD Extraction Modules" are used for defining information from CAD. These modules are used both to describe the site and to define stored in the CAD environment. Although the CAD and simulation modules are separated, the extraction modules are not developed without intimate knowledge of the atomic models. The CAD interfacing and data extraction are integral parts of the described model. The CAD interfacing is developed for the data extraction modules and is described in section 4.4. The data extraction theory is pertinent to the success of CAD-based simulation modeling and is described in section 4.5.

Simulation Environment: The simulation environment contains the simulation atomic model library, and the simulation engine. Before the CAD interfacing is created, the atomic model library should be defined. Atomic models are combined to create the simulation model. The combination of atomic models used to create the simulation system model is defined from CAD. When the simulation is run, the simulation models use the CAD extracted data that is stored in the intelligent data manager.

The development of a CAD-based simulation system can be categorized into four distinct components, as illustrated by Figure 4-5:

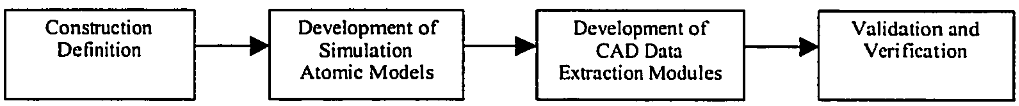


Figure 4-5 Development of CAD-Based Simulation System

1) The first component of the development is the construction definition. The construction processes, the construction site, and the critical factors affecting the construction processes must be identified. The locations of the information defining

was described for the earthmoving example in section 1.2 of this thesis.

2) The second component of the methodology is the creation of 'atomic' simulation models that represent the construction processes. These models mimic the construction activities and are combined to create a system model. The system model uses the information from CAD and simulates the construction. This was developed for the earthmoving example in chapter 3 of this thesis.

3) The third component of the methodology is relating the information available in CAD to the simulation models. The CAD data is extracted using CAD extraction modules, which are developed specifically to relate to the simulation atomic models. These extraction modules can be used to define general site information, specific site data, construction sequencing, construction resources and construction processes for the project structure. This third step is the focus of the following work. CAD provides two main functions: The first is the need to describe the physical site for the simulation through the use of CAD extraction modules. This includes describing the properties of construction objects (or structural components of the project design), described in section 4.3.2. The second focus is on defining the type of simulation model that should be used for the construction of each of these objects, described in section 4.3.3.

4) Verification and validation of the prototype is the final step in the development of a CAD-based simulation system.

The site and sequencing of construction is defined from CAD. The extraction modules are used to describe the information that the simulation model will use regarding the site and the source. For example, in the earthmoving example, the source, roads, placement, number of trucks, type of excavators need to be defined from CAD. The creation of data extraction modules in CAD allows the user to define information for the components of atomic models. With these modules, the user can then be given the ability to define the construction sequence, the type of excavation method, and the excavator that can be used for each block.

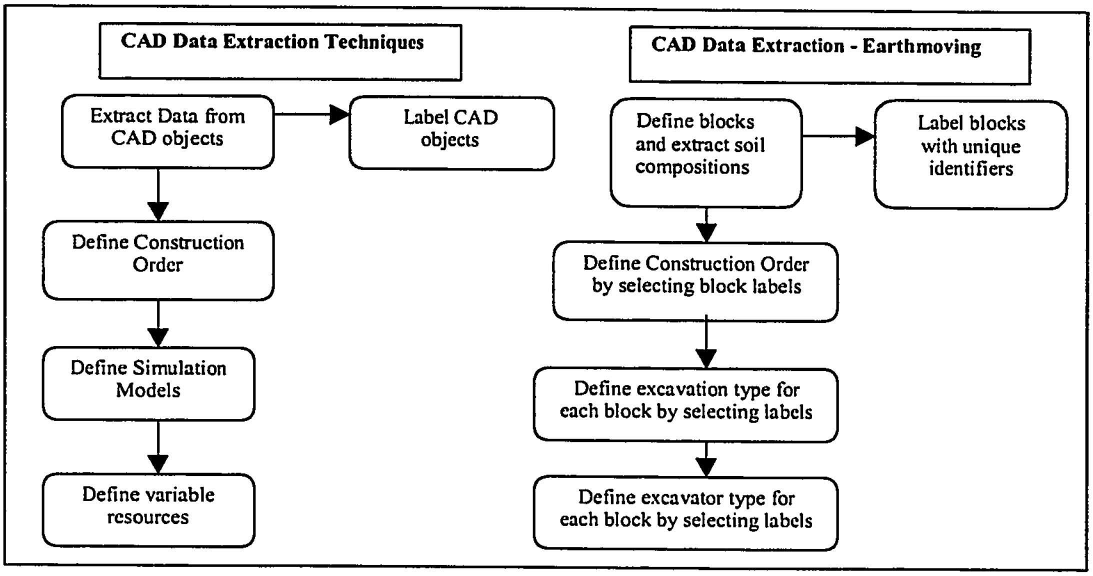


Figure 4-6 CAD-Based Earthmoving Extraction Process

Figure 4-6 illustrates the methods in which the CAD extraction modules are combined to logically define the construction operation. By extracting data for the construction objects in CAD and labeling them, the objects can be referred to for further definition. Labeling involves drawing a unique text descriptor on the construction object in CAD. When the data is extracted, the unique label can be extracted and stored with it. The label is then on the object in CAD. All referencing of the object will then be done by selecting the label, so that all of the information for an object can be related. Using the excavation source as an example, we divide the excavation into blocks and these blocks are labeled. At the same time as the blocks are being labeled, the data for these blocks are extracted and stored with their labels. Once the objects have been labeled, the construction sequence or order is defined by selecting the labels in the desired order of construction. The next step is to define the type of construction process or simulation model to be used for the object, or, in the case of the earthmoving example, the block. The methods of selecting the type of simulation model to be used are discussed in Section 4.3.2. The last step is to define specific resources for each object. Using the earthmoving example, different excavator sizes may be defined for different blocks. By following this simple methodology, construction objects can be defined for the simulation process. This methodology can be used to develop CAD-based simulation systems for any type of construction projects.

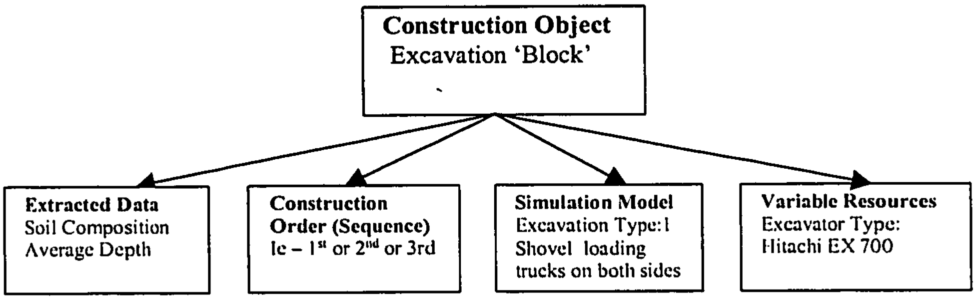


Figure 4-7 Construction Object Properties

model component. For example, from Figure 4-4, the "Roads" CAD-extraction module defines the physical characteristics of the road from CAD. The user defines the road by selecting points along the road, creating a road profile, and then describes characteristics of the road such as rolling resistance and intersections. The Travel-loaded and Travel-unloaded atomic models use physical characteristics such as distances, slopes, resistances, and external traffic parameters. These characteristics affect the velocities of the trucks and thereby the activity durations, cycle times, and overall productivity of the system.

4.3.2 Simulation Models Definition

The goal of this CAD-based simulation modeling methodology is not to create the simulation model library from CAD, but to be able to define the high-level system model from the model library from within CAD. By creating extraction modules that are used to define the simulation model, the user is provided with a method to define the entire simulation model from within CAD.

Three methods can be used for defining the high-level simulation model:

1) The first method involves defining entire simulation model(s) as combinations of atomic models outside of the CAD environment. The user can then define, in CAD, which simulation model is to be used. This method provides an advantage if there are not many different permutations of models.
2) The second method involves defining the combination of atomic models to use within the CAD environment. This method requires that the user maintain a good

there is such variability in the modeling that it is the only logical method.

3) The third method involves defining portions of the simulation model and leaving parts of the model variable. For example, the high level model in Figure 4-8 defines the entire simulation model except the loading model, which is left variable to be defined in CAD.

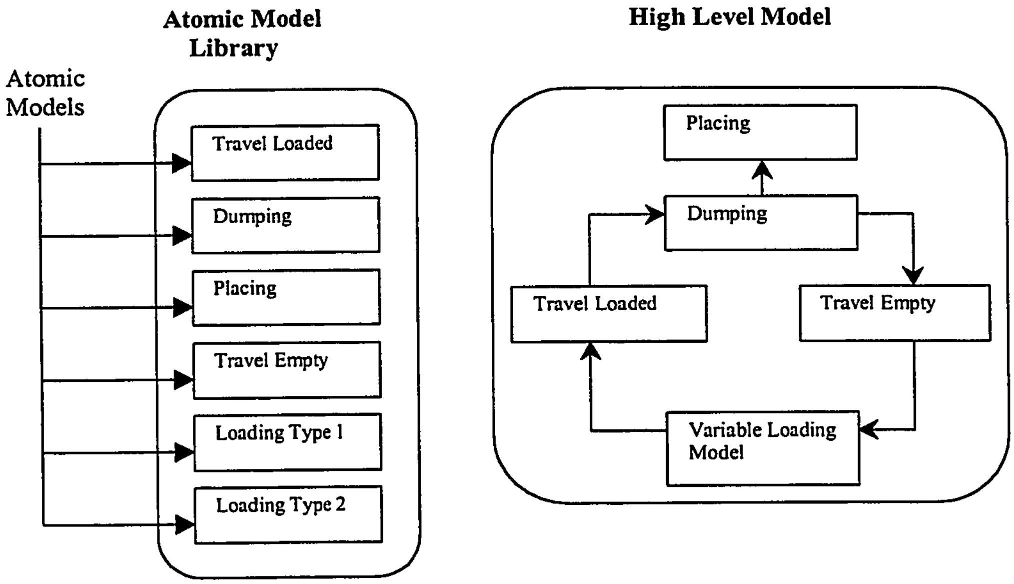


Figure 4-8 Simulation Definition from CAD

The labels of objects in the CAD file are used to describe the high-level simulation model for the construction process. The CAD extraction modules used to define the simulation types must be able to interface with the available types of atomic simulation models. The atomic models should be labeled descriptively so that the user can readily identify the combination of atomic models. The types of atomic simulation models available can be defined in the intelligent data manager. By selecting object labels from CAD and describing the combination of atomic models to use, the high-level through an interface knowledgeable of the atomic models available.

4.4. CAD INTERFACING METHODS

In order to develop CAD extraction modules, a method of interfacing in CAD is required. Since MicroStation is a platform used for design and drafting, it must accept design add-on programs to assist designers in analysis. Thus MicroStation was designed to allow for integration and interfacing with the CAD engine. MicroStation provides three tools for which the user can create interfacing with the internal CAD data: (1) BASIC Macros, (2) OLE Automation – Visual Basic Interfacing and (3) MicroStation Development Environment (MDE).

1) BASIC Macros: The BASIC macros provide the user with a method to quickly and simply extract information; add, modify, or remove elements; and change element attributes. The BASIC Macros allows the user to program in the BASIC language, which is easy to use and is functionally rich.
2) OLE Automation: The OLE (Object Linking and Embedding) Automation allows the user to interface Visual Basic to CAD. The user maintains the same data extraction capabilities as those provided by the BASIC macros. Some of the capabilities of drawing are limited using this type of interfacing.
3) MDE: MicroStation Development Environment (MDE) is an environment for developing MicroStation Development Language (MDL). MDL combines a full set of development tools with direct access to the MicroStation "CAD engine" through built-in functions. MDL is the C programming language executed by MicroStation.

apply. Most add-on programs are developed using MDL. Users are able to interface and control MDL developed add-ons by using any of the three interfacing methods.

BASIC Macros were used for the development of the applications for this thesis as they provide suitable capabilities for the scope of work.

4.5 CAD EXTRACTION METHODOLOGY

Extracting CAD data is the key to successful implementation of CAD-based simulation modeling. There are two main issues in the data extraction theory: the first issue is the question of how can the CAD information be extracted. The second issue is that of what processes are used to extract the data.

4.5.1 Extracting CAD Data

When extracting data using the previously described interfacing methods, the extraction methods must be methodical to ensure that all methods of extraction are covered. The extraction of data can be divided into five methods.

4.5.1.1 Selection Restricted Methods

Before explaining the different selection methods, it is important to discuss the methods of constraining selections. When a select method is used, the search engine looks for elements. Since elements contain attributes and properties, the search engine can be constrained by three methods:

used when the location of an element may be known.

2) Origin – x, y, z co-ordinates of the origin of an element. An origin selection restricted method may be used when the origin of a point is known.

3) Element Attributes; element type, Element Color, Element Level, Element Weight. An element attribute selection method can be used when the user wants to know information about elements containing specific attributes.

These constraining methods allow the search to limit the retrieved data to those desired by the user and to speed up the search engine

.

4.5.1.2 Point Methods – Define by a point.

The user can define a point using the mouse cursor. The Cartesian coordinates of the selected point can be written to a file. Figure 4-9 contains the code for this routine. The point method of data extraction provides the advantage of being element independent. That is, the point can be at any location in the file, including a location containing no elements. The control of the Z coordinate becomes difficult if no element is selected. The point can also be on any location on an element.

```
' Wait for a data point or a reset
MBEGetInpuMBE_DataPointInput, MBE_ResetInput

'If a reset was entered, then exit the
IfnMbEStateInputType=Mbc_ResetInput
   Then    Exit
End If
' Extract the data point coordinates and
Status = MbeState GetInputDataPoint (point,view)
'
X$ = point.x
Y$ = point.y
Z$ = point.z
```

Figure 4-9 MicroStation Basic Point Extraction

Since many objects created in CAD do not possess desired attributes such as volume or area, these attributes can be defined by a combination of points. For example, in defining the topography for a road, the user can select breakpoints in the road to define the road on a point by point basis.

4.5.1.3 Select using a Select Nearest Element

Since CAD can be very congested, a select element routine actually looks for the closest element meeting the constrained requirements. In using the cursor in congested CAD areas, the desired element may not be selected even when you think it is. By using a selection nearest element method, the function will search all elements near the pointer location and restrict the search only to the imposed constraints. This means that even if the user does not select the desired element, the select function will find any elements that are near the selection location and meet the requirements of the constraints. For example, the select method is used to select the labels that define construction objects. It is used to find text of labels in congested areas. In the selection method used for selecting labels, the code searches the area around the pointer for text only, and when text to then write the text to file if it is accepted.

4.5.1.4 Search Entire CAD File

This type of search routine searches the entire CAD file for elements that meet the imposed constraints. In cases where the data required is in known locations and of known types, a search through the entire CAD file can be effective. The search engine can be constrained using any of the constraining methods discussed earlier. This can yield quick, meaningful results. For example, the volumes when calculated, are in the form of text. These text elements are drawn on level 60. In Figure 4-10, the search is restricted to level 60 and for text elements only. The routine involves no selections, it simply extracts all the data from all elements that meet the constraining criteria, text and level 60, in this example. All of the text is then written to file, which can be sorted and searched for the desired data.

```
Sub main
   Dim element as New MbeElement
   Dim filePos as Long
   filePos = element.fromFile (0)
   Do While filePos >= 0
      Call ProcessElement (0, element)
   filePos = element.fromFile (filePos + element.fileSize)
   Loop
End Sub

Sub ProcessElement (nestLevel as Integer, inElem as MbeElement)
Dim gotNext as Integer
   Do
      ' call function that shows info for element at current position
      Call ShowElemInfo (nestLevel, inELem)
      If inElem.isHeader <> 0 Then
         ' if any components in complex, process them recursively
         If inElem.nextComponent = MBE_Success Then
            Call ProcessElement (nestLevel+1, inElem)
         End If
         gotNext = inElem.nextElement
      Else
         gotNext = inElem.nextComponent
      End If
   Loop While gotNext = MBE_Success
End Sub

Sub ShowElemInfo (level as Integer, inElem as MbeElement)
Dim origin as MbePoint
If inElem.type = MBE_text then
   If inElem.level = 60 then
      If inElem.getOrigin (origin) = MBE_Success Then
            print Spc(indent);"Origin :";origin.x; origin.y; origin.z
      End If
   End If
End if
End Sub
```

Figure 4-10 Entire File Data Extraction Search

4.5.1.5 Add-on data processors

External data processors or "add-ons" contain functions that can be used for data extraction. Add-ons developed using MDL language can be controlled using any three of the described interfacing methods. This lets users not only automate MicroStation functions, but also add-on programs. For example, an add-on program "Eagle Point" has a build in data processor that can be harnessed from interfaces to extract point data.

4.5.2 Logical Data Extraction

In cases where the data extraction needs to be entered in a logical fashion for it to be useful, it should be extracted in a defined, ordered fashion. For example, when extracting the data for a road, the data should be extracted in an orderly manner from one end to the other, to ensure continuity and allow distances to be measured between points. Points must be selected in order from the source to the placement. Since the extracted data is used to define the road stationing, or distances between points, the points must be extracted in order for the distances to correspond to actual distances. If the order of the data is not relevant, then logical data extraction is not required.

4.5.3 Process Extraction Methods

Every construction project is unique in that the sequence of events creating the final product may be changed; this will impact the schedule, productivity and timing of subsequence events. In order to create a realistic simulation model, the sequence of construction events must be definable. To define a construction process for a simulation model from within CAD, the sequence of construction must be related directly to the data extracted for the construction objects. Thus, the simulation can directly relate the extracted data to the construction sequence.

The construction process must be defined. Instead of extracting data in the order in which the construction will proceed, thus requiring data to be re-entered for each different process that is simulated, the labeled objects can be used to define the construction sequence. Upon completion of data entry, the construction process can be construction simulation will perform them. The labels will be stored in the construction sequence defined. In order to ensure that no data points are missed, the process routine can change visual attributes of the labels so the user can visually recognize which labels have been defined in the construction process. This process allows the user to test many construction sequences to evaluate the impact on the schedule, resource utilization and productivity, without re-entering all of the data for each object. For example, it was stated earlier that it would be possible to define the sequence of blocks in earthmoving. A process extraction method involves defining the construction sequence of these blocks for the simulation.

4.6 CONCLUSIONS

This Chapter described the CAD-based simulation modeling methodology. The CAD-based methodology does not create atomic simulation models. The methodology requires that a simulation atomic model library be developed prior to the development of the integration with CAD. The methodology allows system models to be developed by combining atomic models from CAD.

CAD extraction modules are developed to define construction objects, construction sequencing, resources, and the simulation models. The CAD extraction modules are developed requiring intimate knowledge of both the atomic models and the construction process. The CAD extraction modules can extract data using any of the various data extraction methods and processes described. The key to the extraction process is defining construction objects and labeling them for further reference.

methodology can be applied to any CAD program that permits the detailed level of interfacing required.

PREDICTING EXCAVATION PRODUCTIVITY USING NEURAL NETWORKS

5.1 INTRODUCTION

The unique environment and variable conditions under which construction projects take place result in great difficulty in accurately estimating productivity. The productivity of a construction operation is defined as the output of the system per unit of time. The estimation of productivity is one of the most important control functions in the construction industry. A number of techniques are used for estimating and analyzing the productivity of construction operations, the most common techniques being analytical. Another technique for estimating productivity is neural networks.

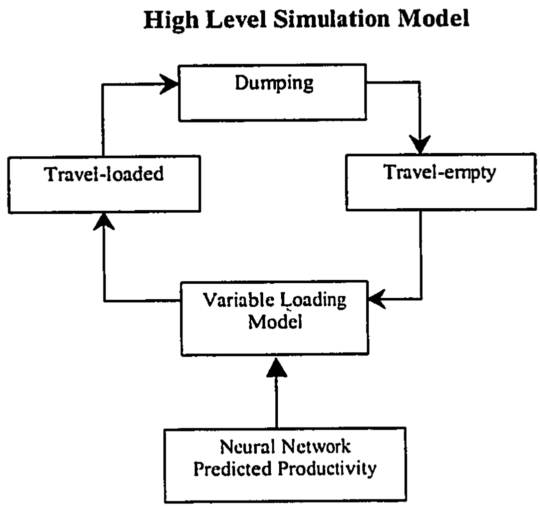


Figure 5-1 Neural Network's Link to Simulation

using neural networks. As illustrated, in Figure 5-1, the neural network developed in this Chapter is designed for incorporation into the simulation models described in Chapter 3.

The production rate of a system is dependent on: (1) predictable factors, (2) unpredictable factors, (3) the interactions between resources on the site, (4) random variation of construction tasks and (5) other external factors.

1) Predictable factors include the structure of the design, which is known. For example, consider earthmoving, where known predictable factors include soil types (where sufficient borehole data exists) and depth of excavations. The known design factors and site conditions, when input into the neural network, combine to predict the output of the network (the predicted productivity) as illustrated in Figure 5-2.
2) Uncertain factors that affect productivity of an operation include weather effects. AbouRizk and Wales (1997) present a method of quantifying weather effects on productivity. This work has not been incorporated into any models in this thesis.

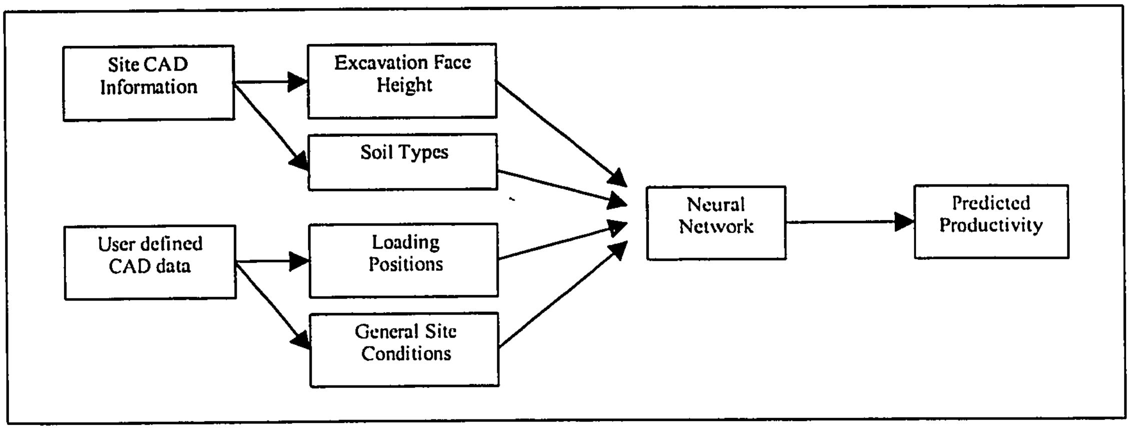


Figure 5-2 Data Flow for Neural Network Productivity Estimation

3) The interactions between resources on the site affect the overall production of the system. This issue has been addressed by linking the neural networks to simulation loading models. The simulation techniques in Chapter 3 provide an effective tool for modeling the interactions between the excavator and trucks, and their effect on productivity.
4) Inherent randomness of construction activity durations is also quantified by using the simulation models. This randomness is quantified by modeling activity durations in the simulation models as random distributions.
5) Labor skills, education, training, and management are also factors that will affect the productivity of a system.

The main reason for using neural networks for construction productivity estimation is the ability of neural networks to perform complex mapping between the various input factors affecting the construction productivity and the output. Other means of analysis, such as regression analysis, lack the means of quantifying many of the qualitative properties involved in the excavation process, such as soil description and soil types. Since these factors cannot be formulated into a regression analysis, a method of analysis allowing descriptive terms must be used. Neural networks allow the user to map these descriptive input variables.

Chao and Skibniewski (1994) provide an example of using neural networks for estimating excavation productivity. They use a robotic desktop excavator for their study. This research extends the idea from the desktop excavator to a practical field application. It presents a method of using neural networks for predicting productivity of excavators performed on an EX-3500 Hitachi shovel excavator and the collected data is used to train and test the proposed model.

5.2 NEURAL NETWORK THEORY

Neural networks are a form of artificial intelligence (AI) that is designed to act in a manner similar to the learning process observed in humans. Neural networks are comprised of multiple simple elements called artificial neurons. These neurons (or nodes) act as artificial individual processors that receive input either directly or from other nodes. These nodes process the input and calculate an output value, which is then sent to the following set of nodes. Most simple neural networks are comprised of an input layer of nodes, a hidden layer of nodes, and an output layer of nodes, as illustrated by Figure 5-3.

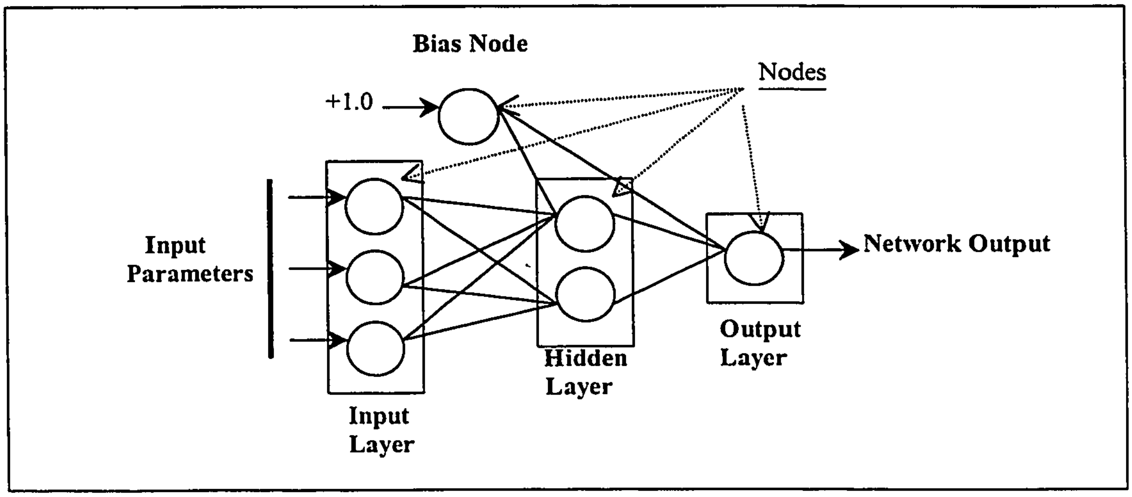


Figure 5-3 Example Neural Network Structure

involves the process of the network observing example data pass through the network, while changing the weighting of input data ,Wjn, into each node's calculation function , F(Z), in an algorithm designed to minimize the total errors resulting at the output nodes. Z is the sum of the input multiplied by their respective weights, and is defined as

$$Z = \sum_{j=1}^{i} X(j) \times W(j)n \quad (5.1)$$

F(Z) is a transfer function defined a mathematical function. The output from each node, n, is Yn = F(Z).

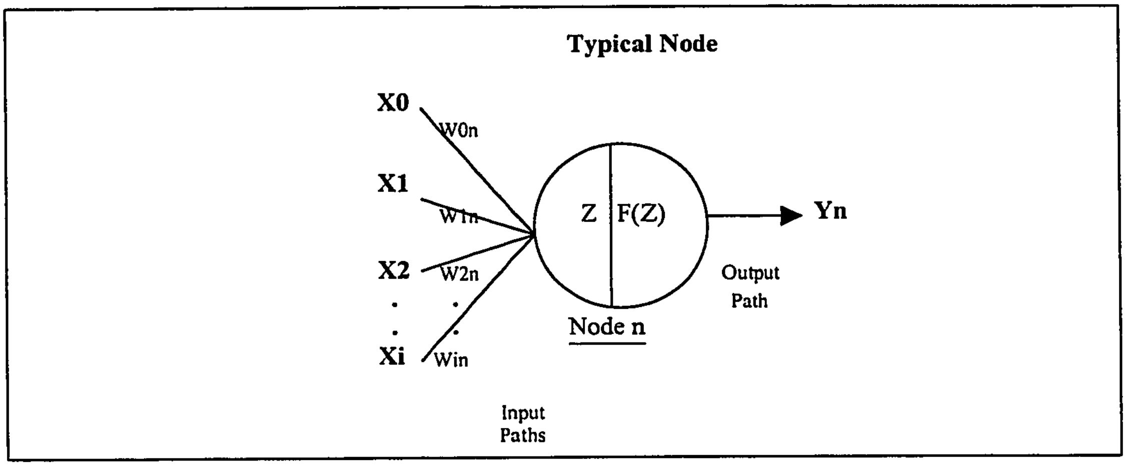


Figure 5-4 Example Neural Network Node

In short, the neural networks learn the relationships that different input have in relation to the output. Thus, after learning, the networks are able to predict output based solely on input parameters, termed recall.

Two basic types of networks are common, supervised or unsupervised. Supervised networks predict output based on the patterns observed in the input and output data that has been used for "learning" or "training." Unsupervised networks are other data sets. The work within this thesis uses supervised networks.

5.3 NETWORK DEVELOPMENT

The development of the neural networks for excavator production estimation is described by 5 components: (1) Problem definition, (2) Data Collection, (3) Network development, (4) Network training and, (5) Network testing and validation.

5.3.1 Problem Definition

The goal of this work was to be able to predict the productivity of a particular excavator based on the input soil conditions, the loading posisiton, and some general descriptive site characteristics. Neural networks provide an effective tool for evaluating relationships between input parameters of the physical site conditions and the resulting productivity output.

5.3.2 Data Collection

To test the applicability of using neural networks for excavation productivity, an excavator which is monitored carefully to provide the accurate data was chosen. It was decided to collect data on an EX-3500 Hitachi Shovel. The EX-3500 has a 24 cubic yard bucket and its historical production ranges from 800 to 1700 bank cubic meters (BCM) per hour. A bank cubic meter is a measure of the earth before it is excavated as it lies in its natural state. The excavator always has a foreman allocated solely to monitor its operation. This foreman performed the data collection.

informal survey of project managers, superintendent and foreman was completed to determine the data that affects the production and the information that should be collected to most accurately reflect the factors affecting the production. The survey determined that the factors affecting production includes the types of soils excavated, general site conditions, loading equipment utilization, and the relative positioning of the excavator and trucks. The results of the survey resulted in the development of the "Excavator Productivity Data Collection Sheet" displayed in Figure 5-5.

The following method was used for data collection: At the end of every shift, the foreman in charge of the excavator filled in the data collection sheet. Data was collected over a three-month period from June 1997 to August 1997. A total of 132 production sheets were collected over this period. At the time that data collection began, the project for which data was collected from, had been in progress for over four months.

The goal of the production portion of the data collection was to obtain an 'ideal production'. The ideal production is defined as the production that the excavator would obtain if it never had to wait for trucks, required no repositioning or maintenance, and took no breaks. In other terms, the ideal production is obtained if the excavator is loading 100 percent of the time. The goal of determining the ideal production for each shift allows us to compare the results of changing soil and site conditions on the productivity of the excavator. It also allows the effects of interactions between trucks and the excavator to be determined through simulation.

Project: [ ] Shift: Day ☐ Night ☐

Date: [ ] Excavator: [ ]

Soil Type | **Soil Conditions** | **Excavator Footings**

Sand ☐	Clay ☐	Soft ☐	Wet ☐
Silt ☐	Rock ☐	Hard ☐	Dry ☐
	Overburden ☐	Frozen ☐	

Loading Position

Approx. Swing Angle: [ ] Working Face Height (m or ft): [ ]

Repositioning Time (min/hr): [ ]

Shovel

Good Side Only ☐
Bad Side Only ☐
Drive By ☐

Production

Excavator Utilization (%): [ ] Loading Duration (hrs): [ ] Total Production (BCM): [ ]

Misc. (This section can be filled out on a weekly basis.)

Bucket Capacity (BCM): [ ] Soil Density: [ ]

Excavator Condition:

Excellent ☐ Fair ☐
Good ☐ Poor ☐

Figure 5-5 Excavator Productivity Data Collection Sheet

production. The total shift production is defined as:

$$Total\ Shift\ Production\ (BCM) = \sum_{i=1}^{j} Number\ of\ Loads(i) \times Truck\ Capacity(i) \quad (5.2)$$

where j is the number of trucks in the system. By comparing the number of loads to actual surveyed quantities, the actual truck capacities had been accurately calculated, and were known prior to the start of data collection.

The average utilization is defined as:

$$Avg.\ Utilization\ (\%) = Excavator\ Utilization\ (\%) - \frac{Repositioning\ Time}{60} \times 100 \quad (5.3)$$

The ideal production is then a function of the total shift duration, the total shift production, and the utilization of the excavator for loading and is defined as:

$$Ideal\ Productivity\ (BCM/hr) = \frac{Total\ Shift\ Production\ (BCM)}{Shift\ loading\ duration \times Avg.\ Utilization} \quad (5.4)$$

Figure 5-6 illustrates the distribution of the collected ideal productions. The range of productivity is from 1725 BCM per hour to 1125 BCM per hour, a range of 600 BCM per hour. The average productivity is 1421 BCM per hour with a standard deviation of 125 BCM per hour. A summary of the collected data and ideal productions can be found in Appendix A.

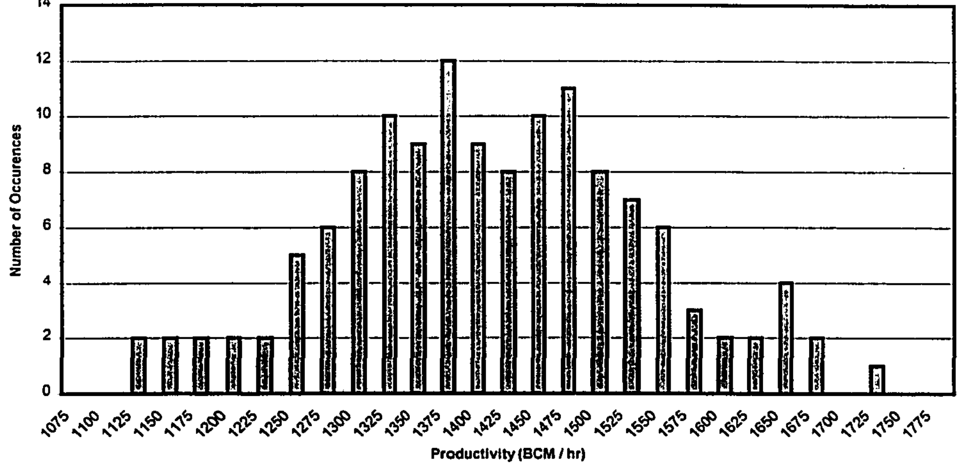


Figure 5-6 Distribution of Collected Production Data

Upon completion of the data collection, the input and output for the proposed network were identified. Four categories of site conditions were identified by construction personnel as factors that affect the ideal production: (1) Soil classification, (2) General soil description, (3) Excavator footing conditions, and (4) Excavator and truck relative positioning. The foreman collecting data described the input parameters as the average description for the shift.

Data Input:

1) Soil Classification: Five generic types of soil were defined for our data collection; Overburden, silt, sand, clay, and rock. These generic soil types were chosen because they are easy for foremen to identify, and they encompass the entire soil spectrum encountered for this excavator at the location where it was working. Other soil types such as blasted rock, although not encountered at this site, should be included, when encountered. All soil types encountered during the shift are checked.

These descriptions have further effect on the productivity; they describe the soils being excavated. Theoretically, the productivity decreases with frozen or hard conditions. Also included in this group is the excavation face height in meters. Theoretically, the higher the face height, the higher the productivity.

3) Excavator footing conditions: The excavator footing conditions are either wet or dry. Productivity of the excavator is theoretically higher for dry footing conditions.
4) Excavator and truck relative positioning: The excavator truck relative positioning contains four input. The first is the average swing angle. As the swing angle increases, the productivity of the excavator will decrease. The other input is the percent of both side loading, in which trucks are parked, one on each side of the excavator. The percent of good side loading is the percent of the shift during which trucks are loaded on the operator's side of the excavator. The percent of bad side loading is the percent of the shift in which the trucks are loaded on the far side of the operator's side of the excavator. Productivity of the excavator is theoretically highest when both sides are being loaded and the swing angle is minimized.

Data Output: The output parameter is the ideal production defined in equation (5.4). For the remainder of this Chapter, the terms 'productivity' and 'production' shall refer to the ideal productivity defined in equation (5.4).

5.3.3 Proposed Network

Prediction of earthwork excavation productivity uses the neural network shown in Figure 5-7. It contains 13 input nodes, with one hidden layer containing 4 intermediate from the collected data. Other factors such as whether it was a day or night shift, the foreman and the operator were also collected. However, in testing these factors were found not to have a significant impact on the results. The network uses supervised neural networks trained on the collected productivity data.

Generic soil types, general soil description, and excavator footing conditions are all input as zero-one variable. That is, the input value is either one if the input parameter was encountered or zero if it was not. The loading positions are input as percent of shift duration, and the average swing angle is input in degrees. The excavation face height is input in meters. The collected data is entered as the input layer, propagates to the hidden layer via weighted connections and propagates to the outer layer where the productivity is determined. Propagation from layer to layer occurs by summing the node values for the respective connection weights to each node as described by equation 5.1. The activation function used for equation 5.1, in the developed network is:

$$Y(n) = f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \quad \text{.............(5.5)}$$

All network nodes function in a same manner. This activation function used in the excavator neural network is the TanH function that maps values to the range -1 to 1.

The network was determined through the training process. The network consisting of only four hidden nodes is a result of minimizing the number of nodes for the network while maintaining a minimal error. Excessive nodes in a hidden layer results in the network "over-training." Over-training results from the network containing so many options that it can solve each training data point individually without combining factors for the network. The activation function was chosen as the function that trained the four node network to the minimal average error of the output node.

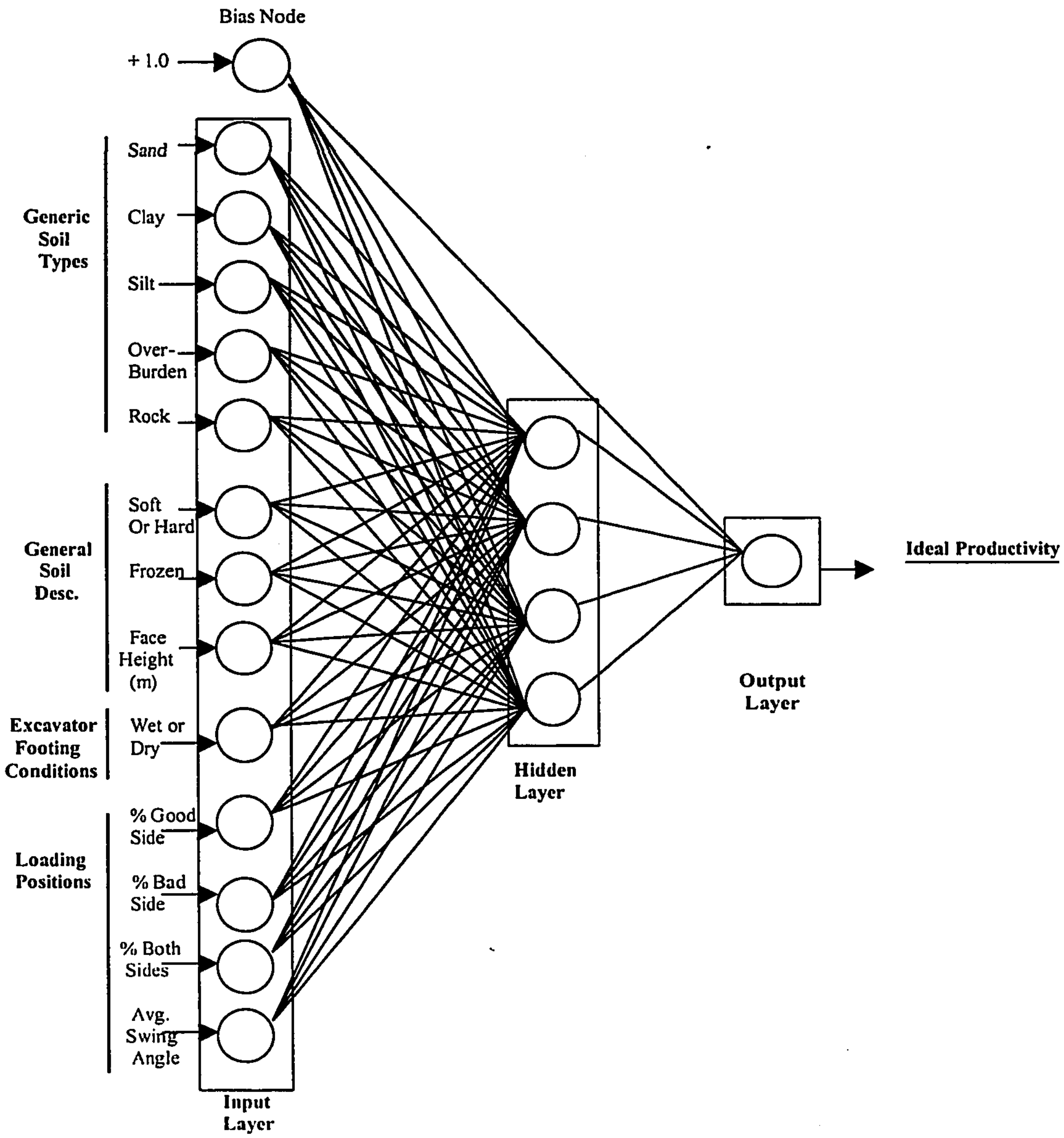


Figure 5-7 Neural Network for Excavator Productivity Estimation

Training the neural network to produce proper results requires a set of data consisting of inputs and the corresponding correct output values. This data is used to train the neural network to deliver the proper output for the given set of inputs. Out of the 132 points collected for the neural network productivity, 20% of the data points were randomly chosen and used for network testing, while the remainder, 80% of the points, were used for data training.

The learning occurs by adjusting the weighted connections between the neurons of each slab such that it minimizes the overall error between the predicted and actual output. During learning, the training data sets used are run through numerous iterations of the learning process, continually training the network towards the minimum achievable error.

During training of the network, the mean absolute error is continually monitored and is indicative of the network performance. At the point where the absolute error levels off, the training is stopped. The result of the learning process is a recall network. The recall network feeds in the input values and predicts the output.

Figure 5-8 displays a sample set of the network predictions for the training data from the recall function plotted against the actual productivity values from the training data set. The network training data is located in Appendix A. The average error between the productivity predicted by the network and the actual productivity is 4.5% absolute and 11.5% over the range of data. This compares to an average 7.3% difference between the actual productivity and the average productivity of 1420 BCM/hr. From the results of was mildly effective.

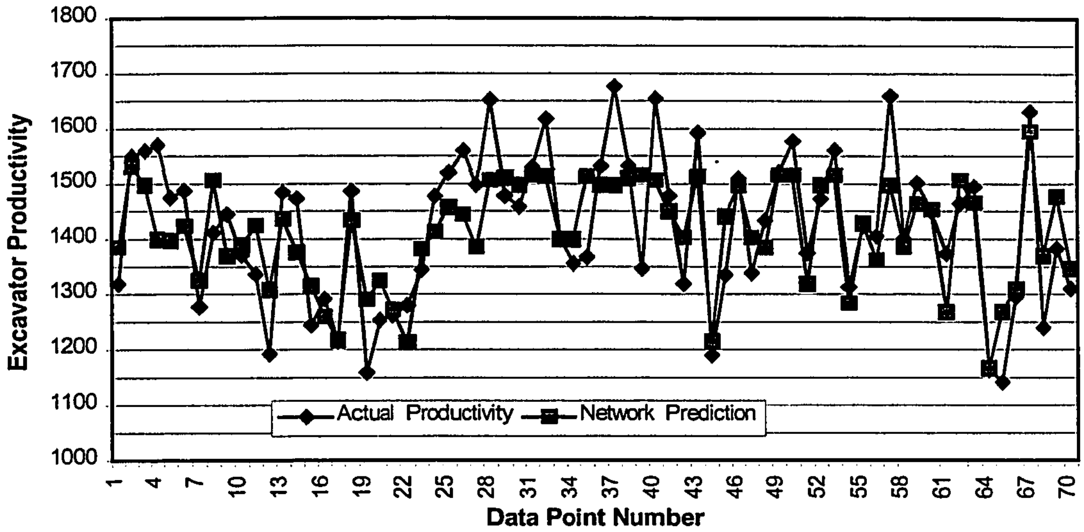


Figure 5-8 Network Training: Network Predictions verse Actual Values

5.3.5 Network Testing

Once the data has been trained, the next step is to test the neural network recall function with data that the network has never encountered. The data points reserved for testing are used in the network to test the recall network for acceptability.

The average error between the productivity predicted by the network and the actual productivity is 5.6% absolute and 15.3% over the range of data. The average variance is 9974. The maximum absolute error is 43% over the range of data and 15% absolute. The difference between the average productivity of 1420 BCM/hr and the actual productivity is 6.4% for the testing data, and the average variance is 12626. A visual inspection of the results in Figure 5-9 indicates that the network successfully estimates the productivity. A comparison between the variances indicates that the neural networks do not significantly improve the prediction of productivity.

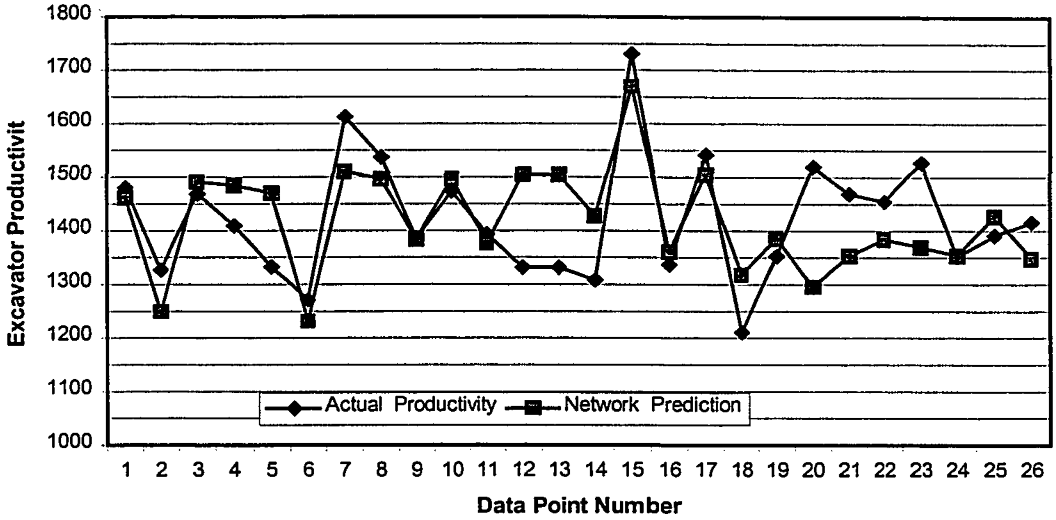


Figure 5-9 Network Testing: Network Predictions verse Actual Values

5.4 CONCLUSIONS

Since the construction environment exerts great influence on construction productivity, site conditions have to be considered. The approach used in this work uses a neural network to map the complex relationships between site conditions and productivity. This approach was unable to prove that neural networks are able to map the complex relationships between the conditions and the productivity of an excavator.

A neural network approach to estimating productivity of an excavator that uses observation data to train neural networks for the prediction ability has been presented.

The results of the network training indicate that the data successfully trained the network to predict productivity based on the observation-input data and the corresponding productivity. Results comparing the test data between the predicted and the actual productivity indicate that the proposed neural network did not significantly improve the prediction of productivity.

5.5 CONFIDENTIALITY

For the purpose of this thesis, all productivity values have been collected from a general earthmoving contractor. In order to maintain confidentiality, all productivity values have been factored by an undisclosed constant value. All productivity values were factored equally so that the full effect of their relationships can still be expressed in this thesis. All productivity values are not, and should not be construed as accurate representations of production capabilities of the Hitachi EX-3500 shovel.

INTEGRATION OF CAD AND ATOMIC MODELS FOR EARTHMOVING: PROTOTYPE DEVELOPMENT AND TESTING

6.1 INTRODUCTION

In order to test the CAD-based simulation modeling approach, prototype development was undertaken. The primary aim of the prototype development and testing was to prove the feasibility of developing CAD-based simulation models. This prototype incorporates the work described in Chapters 3, 4 and 5 and combines them into a special purpose earthmoving CAD-based simulation tool.

The development of a CAD-based simulation system is categorized into four distinct components, as illustrated by Figure 6-1, and described in Chapter 1.

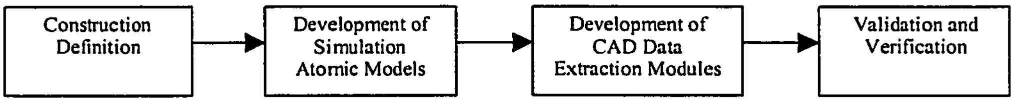


Figure 6-1 Development of CAD-Based Simulation System

Relating the development process to the prototype, the following work applies:

1) The earthmoving site used in the prototype is defined in Figure 6-2. It consists of an excavation source, a single placement area, and a road connecting the two. The excavation site will be divided and defined as described in Section 1.2 of this thesis. Although the placement area will not be defined in any detail, it could be divided and described using subgrids and sequencing in the same manner as the excavation. The

simulation system. It is described in Section 6.2 of this Chapter.

2) The simulation atomic models used are presented in Chapter 3.

3) The development of the CAD extraction modules used for the prototype is detailed in this Chapter.

4) Verification and validation of the prototype, is the final step in the development of a CAD-based simulation system.

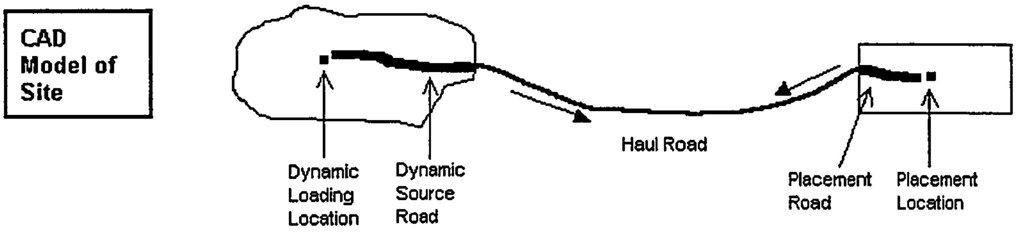


Figure 6-2 Earthmoving Site Description

6.2 EXCAVATION DESIGN USING MICROSTATION AND EAGLE POINT

In order to understand the development of the CAD-based simulation modeling for earthmoving, it is essential to have an understanding of the excavation design and the process in which it is created. The design of the site is in the CAD program MicroStation. In order to design the excavation, an add-on program, Eagle Point is used. Eagle Point is a MicroStation add-on program used for civil engineering and surveying. Eagle Point provides modules for road designs, surface modeling, site designs, data importing and exporting, survey layouts, and water and sewer designs.

The design of the excavation will result in a CAD file that is similar in profile appearance to the site shown in Figure 6-3. To design an excavation, surface elevation data must be imported, surfaces must be created, a final design grade must be defined, intermediate benches must be defined, and finally volumes must be calculated.

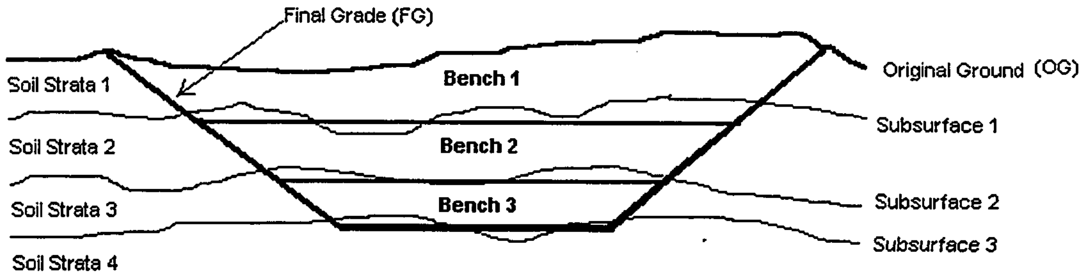


Figure 6-3 Excavation Cross Section

The design of an excavation can be described in five steps:

1) The first step in creating an excavation is to import surface elevation data to develop surfaces in MicroStation that represents original ground and the different soil strata subsurfaces. To import raw elevation data from borehole logs and other sources, the Eagle Point "Data Transfer" function is used.

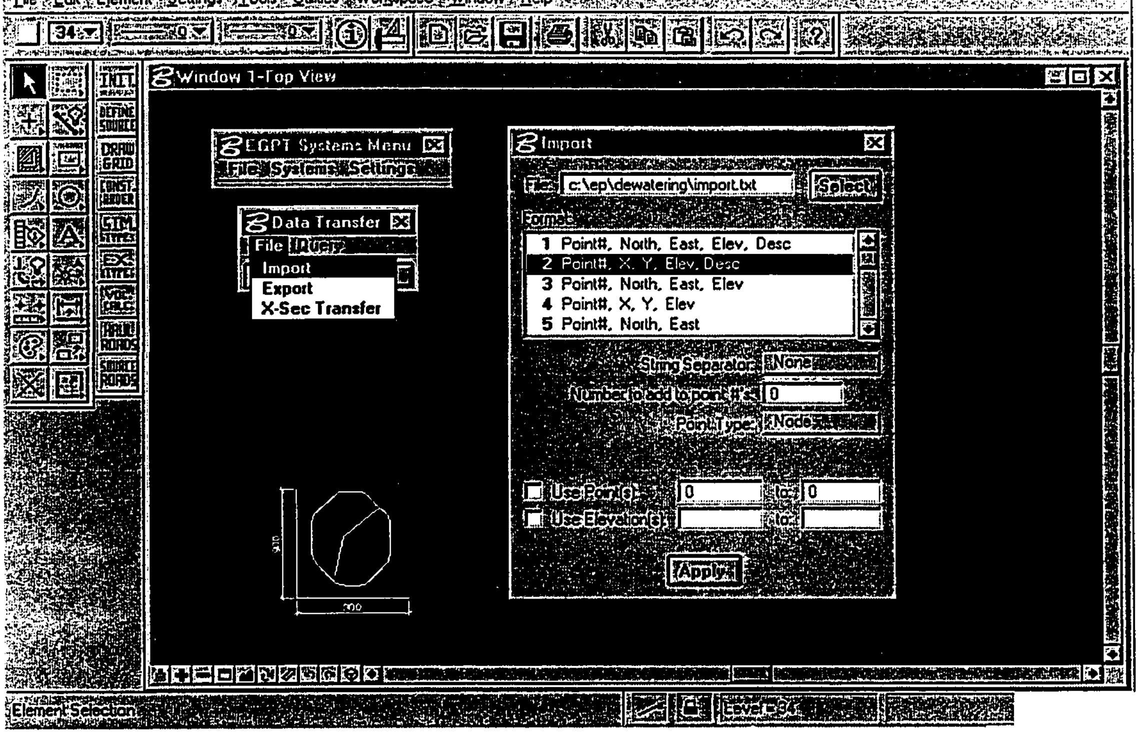


Figure 6-4 MicroStation View: Importing Raw Data using "Data Transfer"

2) The second step is to create the surfaces for the soil strata and the original ground (OG). Surfaces are created using the "Surface Modeling" function within Eagle Point. Surface modeling creates a surface by making a TIN (Triangulated Irregular Network) between points. Each triangle acts as a plane between the points, creating a continuous surface over the area enclosed by the points. Figure 6-5 illustrates the TIN resulting from the points contained for soil strata PF2, from the "Surface Model Library." The TIN is stored on Level 8, while the points used to create the TIN are stored in Level 7. Surfaces are created for each soil strata layer and the OG. When a TIN is created for a surface in the "Surface Model Library," the surface has been created.

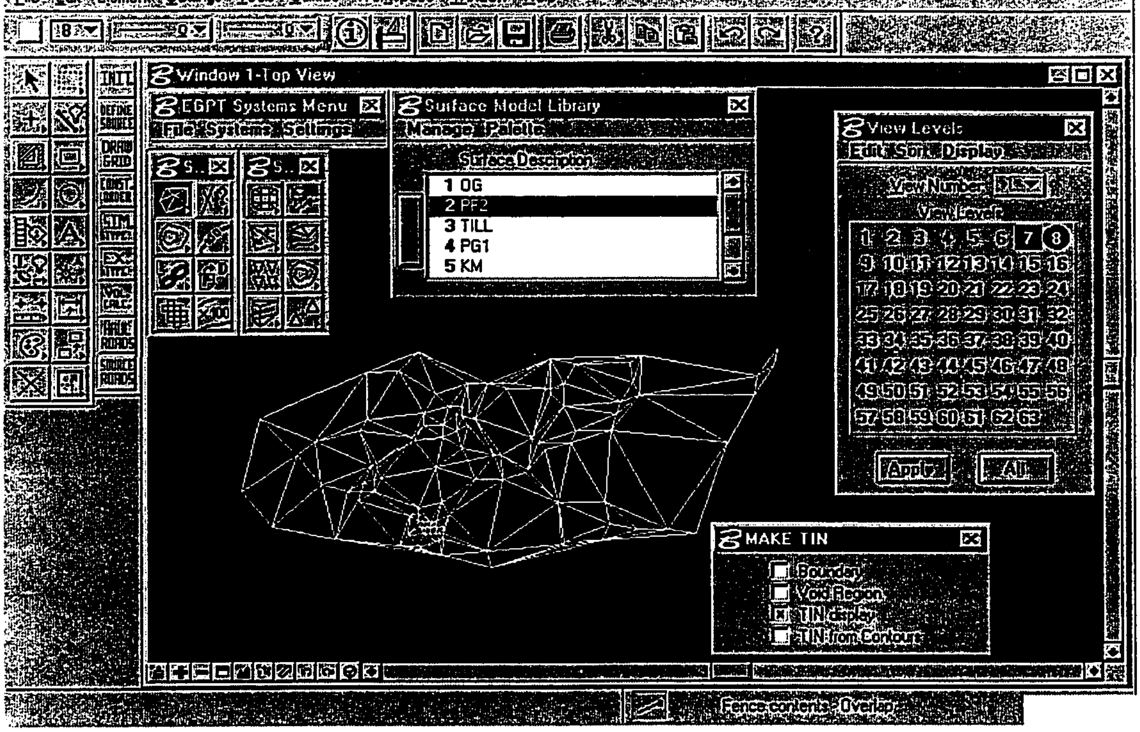


Figure 6-5 MicroStation View: Creating Surfaces using "Surface Modeling"

3) The third step is to create the excavation design, and create a surface for the excavation design, or final grade (FG). The excavation design is typically created from a bottom up approach. The final design of the excavation base and the slopes are known. By drawing the desired excavation base outline, Eagle Point's "Site Design" module can be used to create slope lines from the base outline to the OG. The feature line is the design base line. The slope lines are the lines that extend out from the feature line to the catch line. The catch line is the line that results from the intersection of the slope lines and the highlighted surface, in this case the OG, as illustrated by the "Surface Model Library " in Figure 6-6. The catch, slope and

complete, a surface for the FG is created.

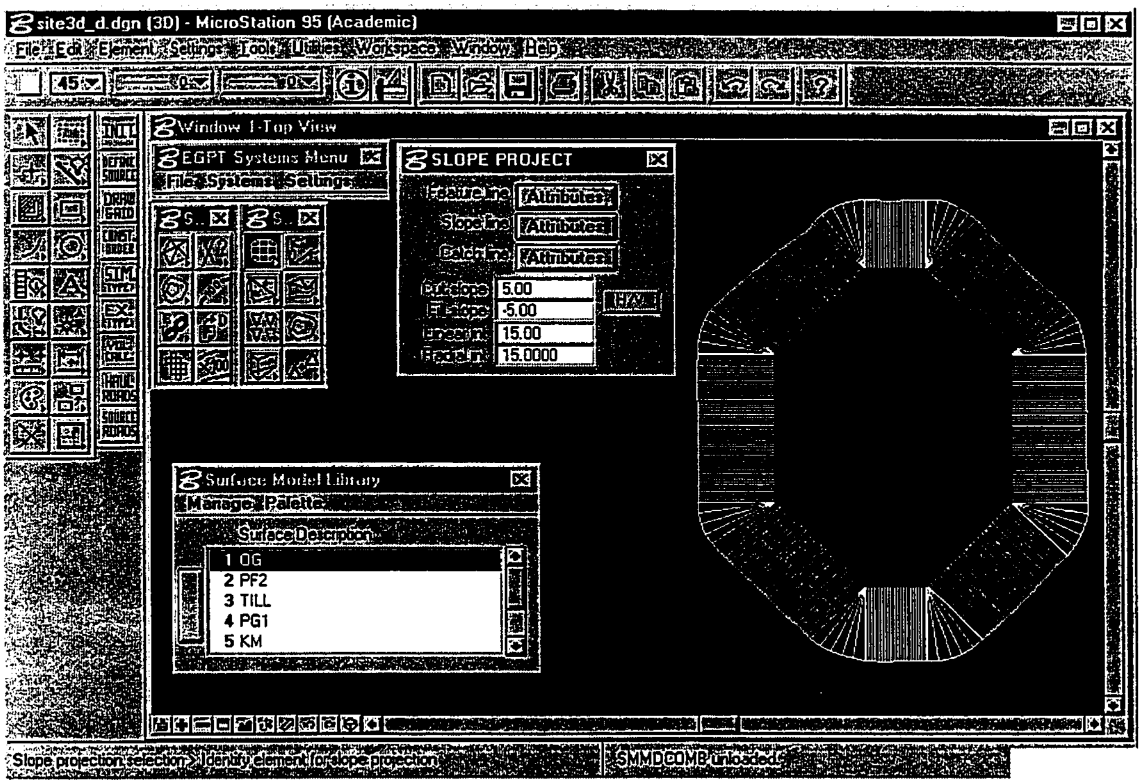


Figure 6-6 MicroStation View: Excavation Design using "Site Design"

4) The forth step is to create intermediate benches for the excavation. Intermediate benches are typically created at a specific elevation. With a flat bench, it is efficient for surveying, leveling and grading. The creation of a flat bench within a design is simple. Using the surface for the FG, a contour is created at the desired bench elevation. This contour is used and linked to the catch line of the FG and OG. The bench surface is developed by creating a TIN as described in step 2. The process is repeated for each intermediate bench.

of information for estimating. The Eagle Point function "Site modeling" provides a volume calculation function as shown in Figure 6-7. In order to determine the composition of soils in each bench, the following procedure is used. Calculate the volumes from the OG and each subsurface to the FG. Then calculate the volumes from the OG and each subsurface to each bench surface. The resulting cut volumes are used to calculate the volume of each soil type for each bench. The volume calculations are drawn as text on Level 60 and are used at a later time.

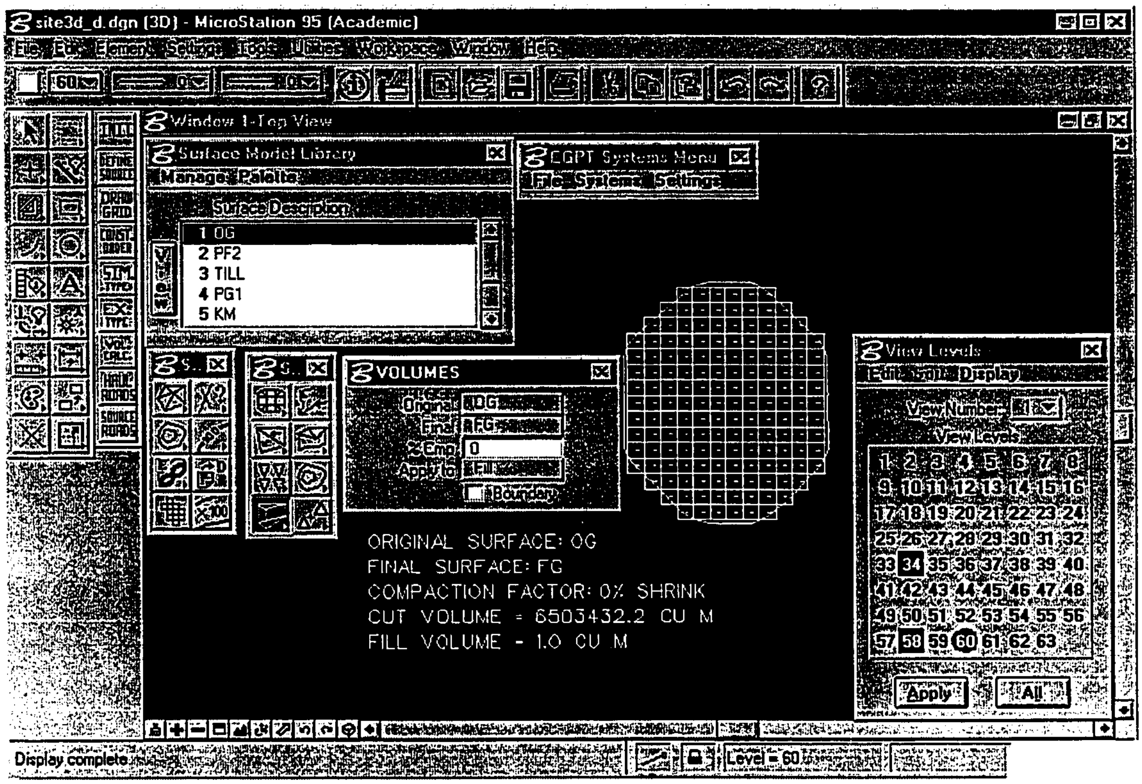


Figure 6-7 MicroStation View: Volume Calculations for Excavation Design

The design of excavations is described above. The CAD design developed, in profile, would appear similar to Figure 6-3 described earlier. This procedure was used to design the excavations used in the following sections.

6.3.1 System Model

The earthmoving CAD-based simulation system is designed as a complete simulation system. The CAD and simulation environments are removed from each other for the purpose of minimizing required simulation knowledge. The atomic models in the simulation model library are described in Chapter 3. The development of CAD extraction modules are described in Chapter 4. The extraction modules for earthmoving are described within this section of the thesis. The neural networks described in Chapter 5 are integrated into the system to estimate excavator productivity. The use of the CAD system requires minimal CAD knowledge to use properly.

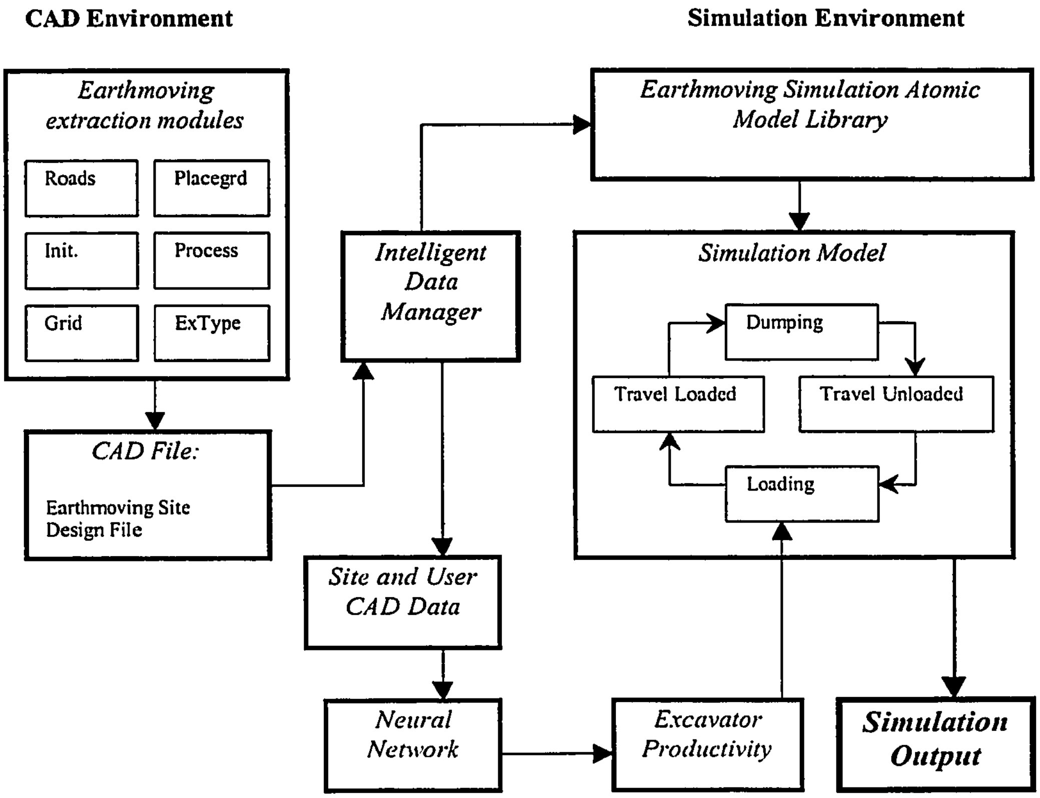


Figure 6-8 Earthmoving CAD-Based Simulation System- Prototype

The simulation output includes cycle times, loading durations, travel times, and resource utilization.

6.3.2 CAD Extraction Modules

The earthmoving CAD extraction modules were created in MicroStation using basic code. The user can customize the interface with specialized toolboxes. For the described system, the "Simulation" toolbox was created to run the macros needed for the site and simulation definition.

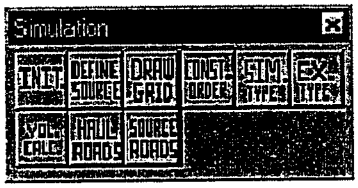


Figure 6-9 Earthmoving Simulation Toolbox

There are nine tools used for the earthmoving CAD-based simulation system:

1) Init. (Initialization)
2) Define Source
3) Draw Grid
4) Const. Order (Construction Order, Sequencing)
5) SimType (Loading Simulation Type)
6) Ex.Type (Excavator Type)
7) Vol. Calc. (Volume Calculations)
8) Haul Roads
9) Source Roads.

Each tool runs a macro that interfaces with the user to define the site for simulation. The nine tools are described in the order that they would be performed, thus mimicking the action of defining the simulation and site in CAD. The definition of the excavation by the extraction modules is illustrated in Figure 6-10.

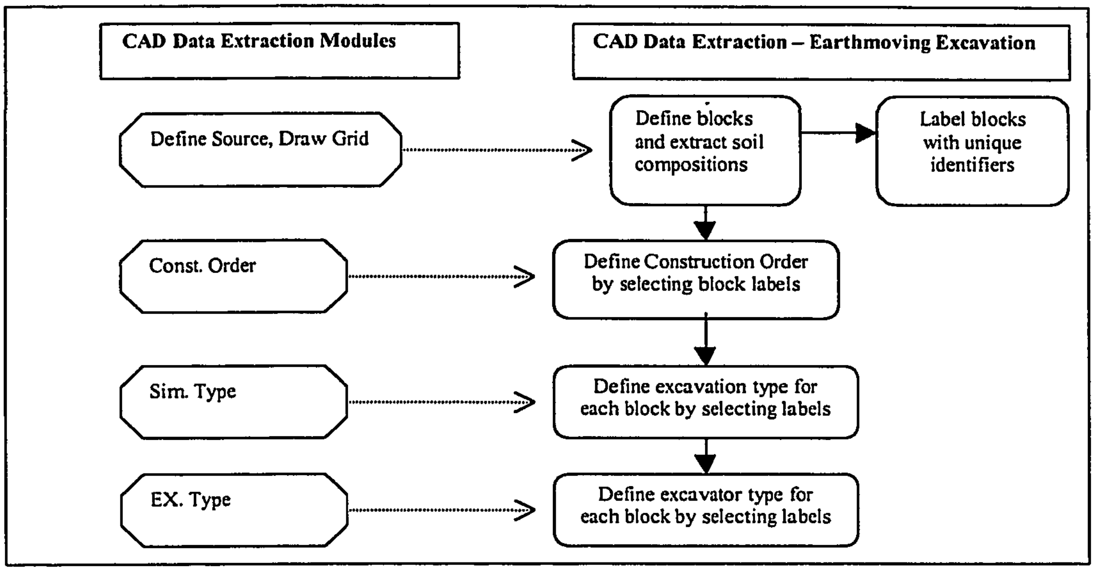


Figure 6-10 CAD Extraction Modules for Defining Excavation

The general characteristics of the site are described using the "Init." tool. First the excavation is broken down into blocks and these blocks are labeled, as discussed in Chapter 1. The "Define Source" and "Draw Grid" tools perform those tasks. Next the construction sequence is defined, using the "Const. Order" tool. The user then defines the simulation type (construction process) that will be simulated on that block, using the "Sim. Type" tool. Finally, the type of excavator to use for each block is defined using the "Ex. Type" tool. Following the definition of the excavation, the roads are defined using the "Haul Roads" tool. Each bench also has a road for the trucks traveling in the excavation. These source roads are defined using the "Source Road" tool.

The actions that the nine tools perform are described below. The actual code for these tools is too in-depth for displaying here, but is found in Appendix B.

Init.: The "Init." tool runs the macro "Initialize". This macro initializes the site for the remainder of the simulation definition. The pseudocode actions of this tool are described in Figure 6-11. First, the number of sources and placements for the site are initialized. Second, the subsurfaces that were developed and are stored in the "Surface Model Library" are defined for simulation purposes. These surfaces are input during the initialization. Finally the benches and a FG for each source are defined.

```
Get Number of Sources
Get Number of Placements
Get OG surface
Get Subsufaces
For Number of Sources
        Get Bench surfaces
        Get FG surface
Next
```

Figure 6-11 Pseudocode for "Init." Tool

Define Source: The "Define Source" tool runs the macro "Grid". This macro defines the source. The pseudocode actions of this tool are described in Figure 6-12. By defining the Southwest and Northeast extremes of the source, this macro creates an imaginary grid around the extent of the source. Centerpoints are created for each subgrid and are labeled and stored to file. The macro then stakes the centerpoints of each subgrid from the original ground (OG) to each subsurface, to each bench, and to the final grade (FG). Since the user has defined the subsurface surfaces, the FG and the benches, no input is

"Site Design". This stake data is all exported to file. With this procedure, the user now has information regarding the composition of the soils for each subgrid. The intelligent data manager determines which subgrids contained a cut between the OG and the FG. The external processor then writes the centerpoints to file "GridtoMS_1.txt".

```
Select Source
User Select Southwest Extent (by Point Method)
User Select Northeast Extent (by Point Method)
Get Subgrid Size (m)
Create Grid: and label
Write Grid to file "GridtoMS_1.txt"
For Number of subsurfaces
        Stake each subgrid from OG to subsurface
Next
For Number of benches
        Stake each subgrid from OG to bench surface
Next
For FG
        Stake each subgrid from OG to FG
Next
Export Stake results to file
```

Figure 6-12 Pseudocode for "Define Source" Tool

<u>Draw Grid:</u> The "Draw Grid" tool runs the macro "PlaceGrid". The pseudocode actions of this tool are described in Figure 6-13. This macro takes the file "GridtoMS_1.txt" and draws a subgrid around each centerpoint. The labels are drawn on an empty level and are used for referencing the blocks. Figure 6-15 illustrates two views of the grid, its subgrids and the labels. View 6 shows the total grid over the source. View 1 shows an enlarged area of the lower left corner of the source, and the labels for each subgrid are clearly visible.

```
For number of subgrids
        If stake on OG to FG is cut
        Draw Subgrid (on level 58)
        Label Centerpoints (with text on level 58)
Next
```

Figure 6-13 Pseudocode for "Draw Grid" Tool

__Const. Order:__ The "Const. Order" tool runs the macro "Process". The pseudocode actions of this tool are described in Figure 6-14. This macro allows the user to define the sequence in which the blocks are excavated. The user views the source design and the subgrids, with the labels attached, as in Figure 6-15. Using a selection function the user selects labels in the order that the construction will take place. The selection function writes the label to file "ConstructionProcess_1.txt". Once a label has been selected, the label changes color, so that the user is aware that it has been included in the construction sequencing.

```
' User selects label in desired construction sequence
For Number of subgrids
        User Select label (using Select Nearest Element Method)
        Write Label, order to file
        Change label color
Next
```

Figure 6-14 Pseudocode for "Const. Order" Tool

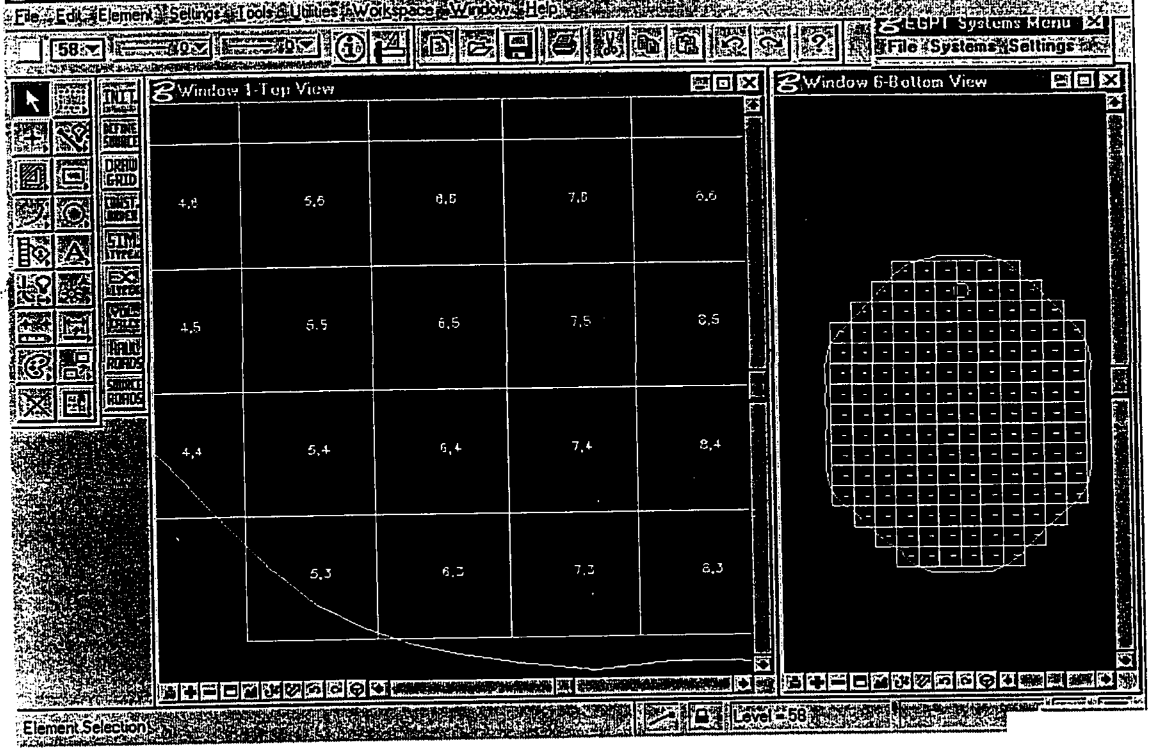


Figure 6-15 MicroStation View: Excavation Grid

__Sim. Type:__ The "Sim. Type" tool runs the macro "Simulation Type". The pseudocode actions of this tool are described in Figure 6-16. This macro allows the user to define the type of excavation simulation (from Chapter 3) that will be executed for that block. The user selects block labels for each simulation type. The selected labels are written to "ExcavationType_1.txt" along with the excavation type chosen for each subgrid.

```
' User selects loading simulation type
For Number of subgrids
        User Select label (using Select Nearest Element Method)
        Choose loading type
        Write Label, loading type
        Change label color
Next
```

Figure 6-16 Pseudocode for "Sim. Type" Tool

this tool are described in Figure 6-17. This macro allows the user to define which excavator will be used for each subgrid. The selected labels are written to "ExcavatorType_1.txt" along with the excavator chosen for each subgrid. If available, different excavators may be used for different locations in the source. For example, smaller excavator would be used at 2 to 3 meter cuts than at 8 to 9 meter cuts.

```
' User selects excavator type
For Number of subgrids
        User Select label (using Select Nearest Element Method)
        Choose excavator type
        Write Label, excavator type
        Change label color
Next
```

Figure 6-17 Pseudocode for "EX. Type" Tool

<u>**Vol. Calc.:**</u> The "Vol. Calc." tool runs the macro "Volumes". The pseudocode actions of this tool are described in Figure 6-18. This macro extracts volumes from CAD and writes them to file. Eagle Point "Site Modeling" contains a volume computational function that computes the cuts and fills between two surfaces. The computed volumes are created into text, and can be placed in the CAD design file. By placing only this volume computational text on Level 60, this macro can extract all volume computational text to a file. This data is used for validation of the subgrid volumes, to ensure the staking performed by the macro "Grid" worked correctly.

Search Entire CAD File
Constrain by Level 60
Constrain by Element type = "Text"
Write "Text" to file
Sort "Text" for Cut volumes

Figure 6-18 Pseudocode for "Vol. Calc." Tool

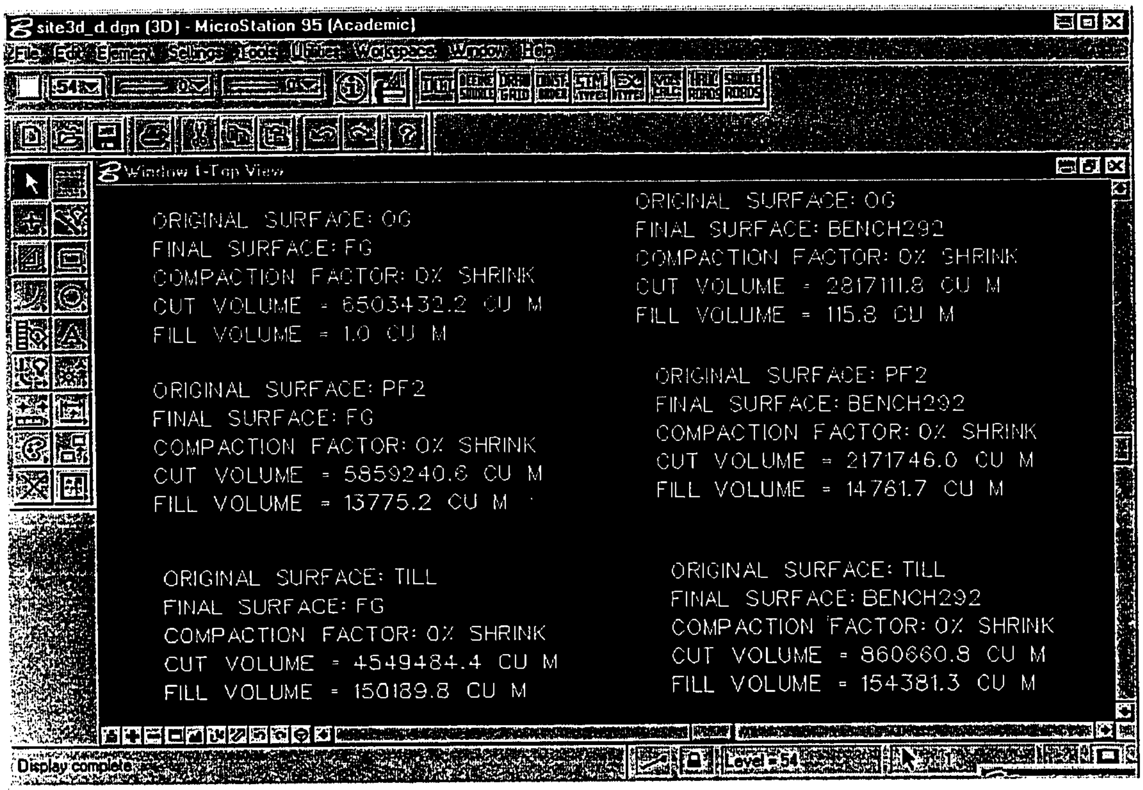


Figure 6-19 MicroStation View: Calculated Volumes in Text

Haul Roads: The "Haul Roads" tool runs the macro "HaulRoads". The pseudocode actions of this tool are described in Figure 6-20. This macro allows the user to define the haul roads on a section by section basis using a point method of data extraction. The road is defined through a series of points that capture any changes in the continuity in the road. Defining points occur where the rolling resistance changes, the slope changes, there is an intersection, or if there is a corner in the road. Figure 6-21 shows the haul road. The small "x"s along the haul road are labels that occur at the break points. The text "Int" in the figure shows the location of the intersection. The road contains 2 non-dimensionally driven properties: (1) rolling resistance and (2) the intersection status. The rolling resistance and intersection properties are defined through a user interface at the end of each road section. The intersection status is zero if there is no intersection at the end of the road section. If an intersection exists at the end of a road section, an intersection identification number is used. The rolling resistance of the road is a function of the energy of the truck absorbed by the road as a percentage. A rolling resistance of 2% is characteristic of a hard, well-packed road, and a rolling resistance of 8% is characteristic of a rutted soft dirt roadway.

```
Select Source
Select Placement
For Number of Road Sections
        Get Road Section Start Point (by point method)
        Get Road Section End Point (by point method)
        Get Rolling Resistance
        Get Intersection Status
        If intersection = yes then Get Intersection number
Next
```

Figure 6-20 Pseudocode for "Haul Roads" Tool

Table 6-1 shows the actual output from the road shown in Figure 6-21. The road is defined by a sequence of points. Point 1 to point 2, describes road section #1, point 2 to point 3 describes road section #2, and so on.

				Resistance (%)	Number None = 0
1	469075.1	6352267.9	298.366	0.0	0
2	469626.2	6352686.3	299.000	3.0	0
3	470226.5	6352962.5	299.000	3.0	0
4	471181.3	6353237.5	298.000	2.5	0
5	471664.5	6353285.5	300.000	2.5	0
6	472096.0	6353187.5	300.000	2.5	0
7	473134.1	6352787.5	295.000	3.0	1
8	473750.0	6352890.6	293.000	3.0	0
9	473966.6	6353040.4	290.000	3.0	0

Table 6-1 Road Output: Haul Road HR_1_1

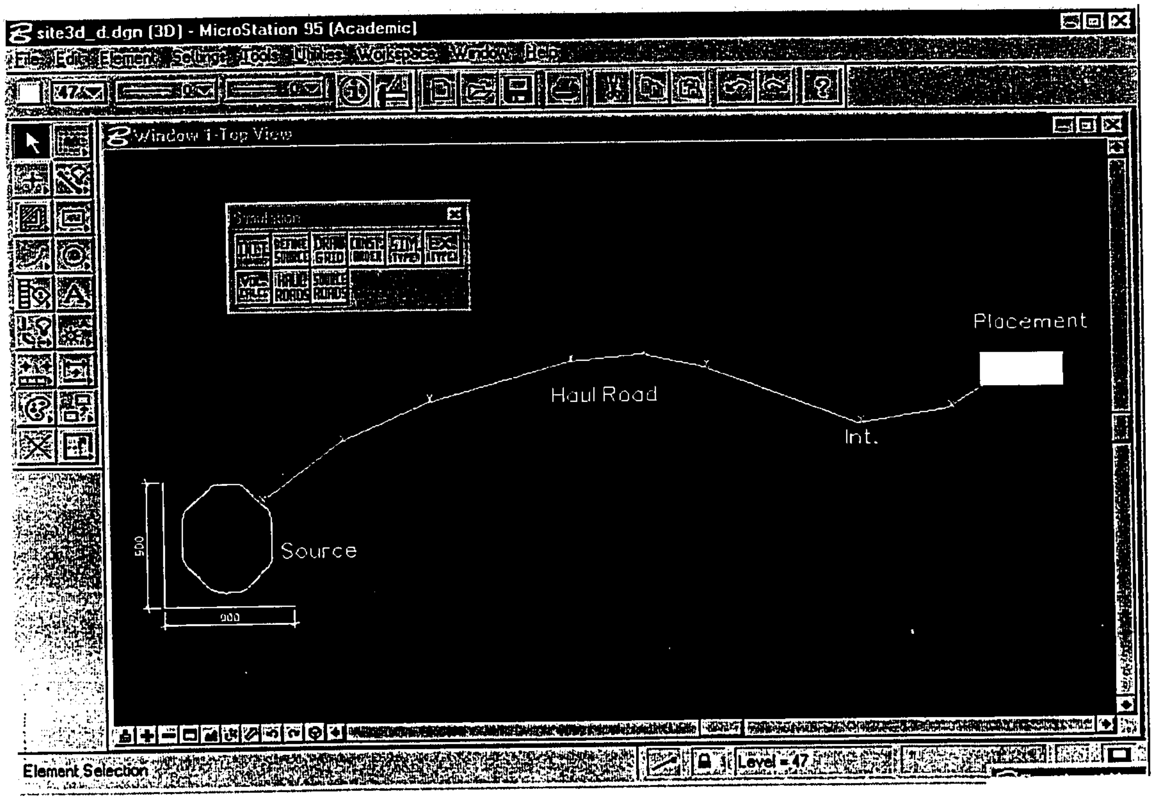


Figure 6-21 MicroStation View: Haul Road

Source Roads: The "Source Roads" tool runs the macro "SourceRoad". The pseudocode actions of this tool are described in Figure 6-22. Since the location of the excavator in the excavation changes, the distances that the trucks need to travel in the source changes. The source road is used by trucks for traveling in the excavation. Each bench has its own source road. The trucks travel from the excavator to the nearest point on the road and then follow the source road to exit the source. This macro allows the user to define the main haul road within each source for each bench, section by section. Source roads contain only one non-dimensional property, rolling resistance. The rolling resistance of the road is entered at the end of each section through a user interface. The small "x"s describe points of break in the road, as in "Haul Roads" tool. Each bench for each source contains a source road. The source roads are defined from the haul road towards the interior of the source. The source roads are defined in sequence from the source to the placement. From the extracted data, strait lines are formed between neighboring points, allowing the calculation of distances and slopes.

```
Select Source
Select Bench
For Number of Road Sections
        Get Road Section Start Point (by point method)
        Get Road Section End Point (by point method)
        Get Rolling Resistance
Next
```

Figure 6-22 Pseudocode for "Source Roads" Tool

				Resistance (%)
1	469075.1	6352267.9	298.366	0.0
2	469012.1	6352221.7	291.000	4.0
3	468778.4	6352026.4	291.000	4.0
4	468720.2	6351711.7	291.000	4.0

Table 6-2 Source Road Data: Bench 1

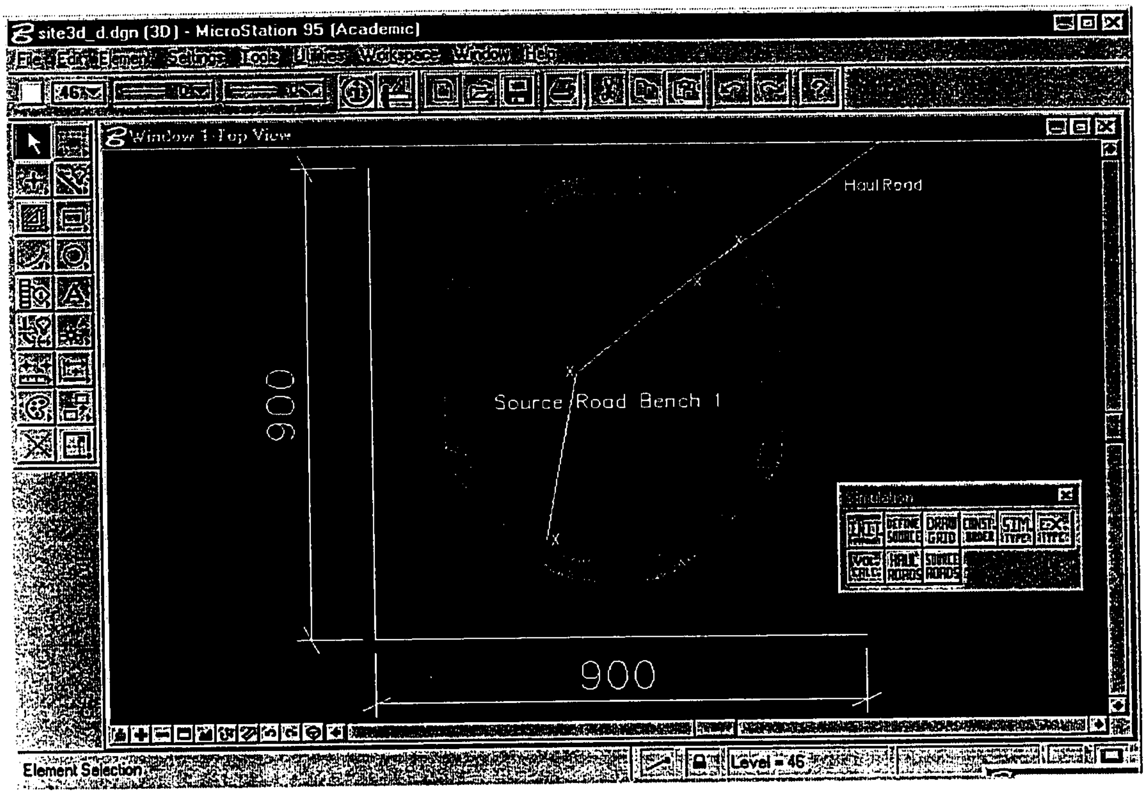


Figure 6-23 MicroStation View: Source Road Bench 1

Verification is the process of ensuring that the model matches the simulation analyst's concept of the system. The following CAD functions require verification:

1) The haul road is described on a point by point basis. Verification that the roads are correctly defined is required. Defined properties requiring verification include the calculated slopes, distances, velocities of the trucks, and subsequently the travel times.
2) The source roads and travel functions need to be verified. It is important to ensure that the method of calculating distances in the source is working according to the design.
3) Other CAD extraction models define the simulation type, construction sequencing, and excavator type. It is important to ensure that these are working correctly.
4) The calculations of volumes, and staking also require verification and validation.

6.4.1 Verification of Haul Roads and Velocities

Haul roads are defined using the Haul Road tool. The haul road tool simply writes points to file. In order to define a haul road, the distances and slopes are extracted from these points. A rolling resistance, and an intersection property, is also defined for each road section. The raw data extracted from CAD for a road is displayed in Table 6-1. Table 6-3 describes the road developed from the raw data in Table 6-1. The section length is defined as the distance between the road section start point and the road section end point.

$$Distance\ (m) = [(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2]^{1/2} \qquad (6.4)$$

Slope (%)

Distance

Road Section	Road Section Start Point			Road Section End Point			Station	Length	Slope %
	X_1	Y_1	Z_1	X_2	Y_2	Z_2			
1	469075	6352268	298.37	469626	6352687	299.00	691.9	691.90	0.09
2	469626	6352687	299.00	470227	6352963	299.00	1352.7	660.80	0.00
3	470227	6352963	299.00	471181	6353238	298.00	2346.3	993.60	-0.10
4	471181	6353238	298.00	471665	6353286	300.00	2831.9	485.60	0.41
5	471665	6353286	300.00	472096	6353188	300.00	3274.4	442.50	0.00
6	472096	6353188	300.00	473134	6352788	295.00	4386.9	1112.50	-0.45
7	473134	6352788	295.00	473750	6352891	293.00	5011.4	624.50	-0.32
8	473750	6352891	293.00	473967	6353041	290.00	5274.8	263.40	-1.14

Table 6-3 Road Properties (1)

Table 6-3 verifies that the distances computed by the CAD road routine accurately reflect the distances between points. Since the velocities of the trucks are dependent on the total resistance, the equations used to determine velocities must be verified to ensure that they accurately reflect the desired velocities.

Road Section	Slope (%)	Rolling Resistance (%)	Total Resistance Loaded	Total Resistance Unloaded	Intersection Number	Velocity Loaded km/hr	Velocity Unloaded Km/hr
1	0.09	3	3.09	2.91	0	36.9	53.3
2	0.00	3	3.00	3.00	0	37.6	53.2
3	-0.10	2.5	2.40	2.60	0	41.8	53.9
4	0.41	2.5	2.91	2.09	0	38.2	54.7
5	0.00	2.5	2.50	2.50	0	41.1	54.0
6	-0.45	3	2.55	3.45	1	40.8	52.3
7	-0.32	3	2.68	3.32	0	39.9	52.5
8	-1.14	3	1.86	4.14	0	45.4	50.7

Table 6-4 Road Properties (2)

The Caterpillar Handbook (1997) presents a method for determining velocities of trucks. By incorporating the total resistance where against the truck, and the weight of the truck. The method presented by Caterpillar does not lend itself to an automated prediction of velocities from input parameters. Each time a new total resistance is encountered, the user is required to extract the velocity from the charts. A simplified method of velocity determination has been developed for this work. By extracting the velocities from the Cat handbook, for a range of total resistances, a velocity prediction equation is developed:

$$\textit{Predicted Velocity} = Sf \times Exp\,((-1 * (TR - Mean)\ \hat{}\ 2) / StDev) + Yf \quad \ldots\ldots\ldots\ldots\ldots\ldots (6.7)$$

where TR is the total resistance, and Sf, Mean, StDev and Yf are constants. Solving the constants in equation (6.7) is performed using an iterative solver and the results are shown in Table 6-5. The error is the difference between the predicted velocity and the velocity from the Cat handbook. The predicted velocity is obtained using equation (6.7). The solver's objective function is to minimize the total error by changing the four input variables. The resulting equation factors, corresponding to the minimal total error, are shown in Table 6-5.

Using equation (6.7) with the constants from Table 6-5, the velocity of an unloaded 793 truck can be determined for any total resistance value. Figure 6-24 compares equation (6.7) with the velocities obtained from the Caterpillar charts. Through visual inspection of this figure, equation (6.6) can be confirmed as a valid estimate of the truck velocity.

Total Resistance (%)	Velocity from Caterpillar Charts	Velocity predicted from normalized distribution	Diff
-15	22.00	21.80	0.04
-12	30.00	29.45	0.30
-8	41.00	41.89	0.80
-6	50.00	47.84	4.67
-3	53.00	54.41	1.98
0	54.00	56.40	5.78
2	54.00	54.79	0.63
4	53.00	51.07	3.74
6	51.00	45.76	27.48
8	38.00	39.58	2.50
10	30.00	33.26	10.61
15	20.00	20.28	0.08
20	14.00	13.47	0.28
			58.88 Total Diff^2

Scale Factor (Sf)	46.06
Yfactor (Yf)	10.38
Mean	-0.37
StDev	153.52

Table 6-5 Truck Velocity Calculations

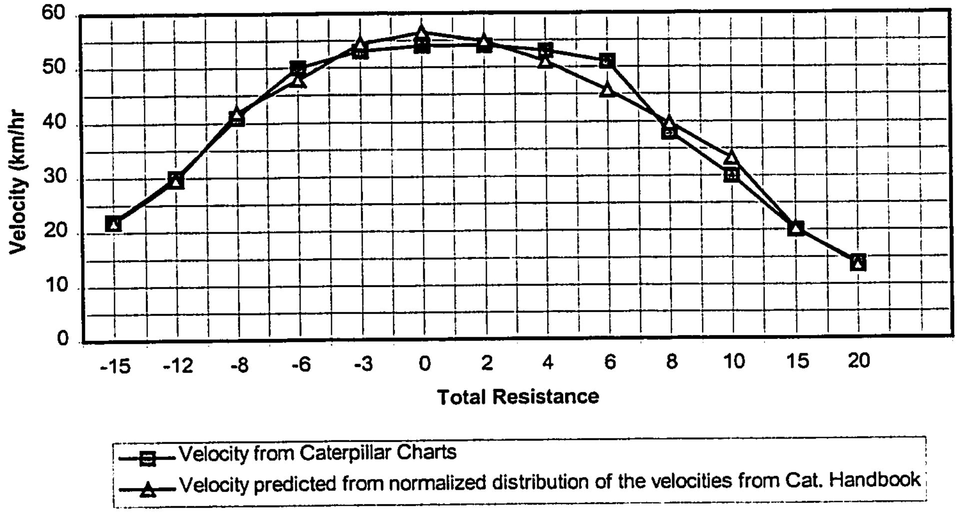


Figure 6-24 Velocity Profiles of an Empty 793 Truck

6.4.2 Verification of Source Roads

The source roads use the same method of determining velocity, distance, and slope as the haul roads. In addition to the distance determined from the source roads, an additional distance function is added. The source roads use a function to determine the distance from the center of the subgrid to the nearest source road for the current bench. The function then determines the remainder of the distance that needs to be traveled on the source road to exit the source. Verification of this source roads function can be performed through a trial approach. One subgrid shall be selected and the resulting path shall be analyzed to conform to the intent of the designer.

Figure 6-25 illustrates the actual site for the point used for verification. Table 6-6 is the results of the query performed by the simulation travel loaded to determine the path of the truck in the source. The large "X" illustrates the location of the point of excavation. From the excavator, the truck then travels to the source road defined by the white line. The distance from the excavator to the source road, as indicated in table 6-5, is 217.4 meters. Then the remaining 270.7 meters of leg 2 of the source road is traveled, followed by leg 1. In considering the scale on the figure, the definition of source roads can be verified by visual inspection.

Arterial Section	Length	Slope	Rolling Resistance	Distance	Leg Distance	Leg
1	78.47	9.39	4	0	78.5	1
2	304.6	0.00	4	217.4	270.7	2

Table 6-6 Source Road Definition: Verification

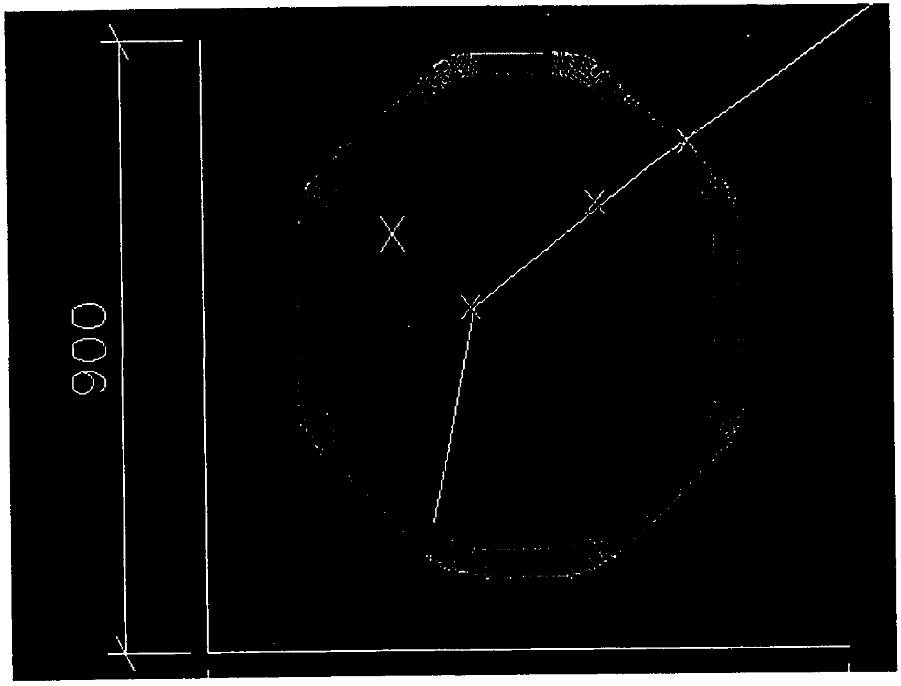


Figure 6-25 MicroStation View: Source Road Verification

6.4.3 Verification of Construction Order

Verification of the construction order, which is defined by the user, must ensure that the sequence of construction must follow the order defined by the user, and the inherent logic built in to the intelligent data manager. The direct input from CAD defining the construction sequence is used by the intelligent data manager to create an "Active Order." The active order is the order in which the construction process is simulated. Table 6-7 illustrates a sample set of data extracted directly from the intelligent data manager. The logical order follows that Bench 1 is excavated, followed by Bench 2 and Bench 3. Many labeled subgrids in Bench 2 and 3 are not part of the "Active Order" because when the subgrid is extended down from the first bench, it falls outside the excavation limits in the lower benches. By examining the "Active Order", we see that being followed as intended.

Direct input from CAD

Order	Grid Labels X,Y
1	12,15
2	12,14
3	11,14
4	11,15
5	11,16
6	10,16
7	10,17
8	10,15
9	10,14
10	10,13
11	11,13
12	12,13
13	13,14
14	13,13
15	13,12
16	12,12
17	11,12

Simulation Construction Order

Bench #1				Bench #2				Bench #3			
Order	Active Order	X num	Y num	Order	Active Order	X Num	Y num	Order	Active Order	X num	Y Num
1	1	12	15	1				1			
2	2	12	14	2	157	12	14	2			
3	3	11	14	3	158	11	14	3	271	11	14
4	4	11	15	4	159	11	15	4			
5	5	11	16	5				5			
6	6	10	16	6	160	10	16	6			
7	7	10	17	7				7			
8	8	10	15	8	161	10	15	8	272	10	15
9	9	10	14	9	162	10	14	9	273	10	14
10	10	10	13	10	163	10	13	10	274	10	13
11	11	11	13	11	164	11	13	11	275	11	13
12	12	12	13	12	165	12	13	12	276	12	13
13	13	13	14	13				13			
14	14	13	13	14				14			
15	15	13	12	15				15			
16	16	12	12	16	166	12	12	16	277	12	12
17	17	11	12	17	167	11	12	17	278	11	12

Table 6-7 Construction Order Data

6.4.4 Verification of Grid definition for Volume computations

The grid tool described in 6.2.3.2 breaks the excavation down into small subgrid units so that the user can define the construction process and excavation productivity can be predicted at different locations throughout the excavation. The grid tool is used to stake the different soil surfaces, final grades and benches. Each block therefore has an area and depth for each type of soil.

To verify that the grid tool correctly calculates the volumes, we shall compare the sum of the block volumes calculated for each soil type and compare this result to the traditional method of calculating the total volumes of each soil type.

Traditional volume calculation of the soil composition of an excavation involves calculating the volume between each surface and the excavation FG. The volume of each soil strata is the difference between the volume calculated below the upper soil surface and the volume calculated below the lower soil surface, and is contained in Table 6-8. The traditional method and the sum of the blocks method is shown in Table 6-9.

Surface to Surface Calculations	Calculated Volume
OG – FG	6,503,432
Sub1 – FG	5,859,241
Sub2 – FG	4,549,484
Sub3 – FG	4,044,199
Sub4 – FG	3,184,119

Soil Type	Total: (Traditional)
1	644,192
2	1,309,756
3	505,286
4	860,080
5	3,184,119

Table 6-8 Traditional Volume Calculations

Soil Type	Total (Benches)	Bench 1 Bench292 (Blocks)	Bench 2 Bench283 (Blocks)	Bench 3 FG (Blocks)	Total: Sum of Blocks	Total: (Traditional)	Error (%): Trad. Vs Blocks
1	645,366	645,366	0	0	656,325	644,192	1.88%
2	1,311,085	1,311,085	0	0	1,306,625	1,309,756	-0.24%
3	505,632	505,632	0	0	537,975	505,286	6.47%
4	860,080	226,330	627,230	6,520	844,075	860,080	-1.86%
5	3,184,119	128,699	1,497,576	1,557,844	3,170,375	3,184,119	-0.43%
Total	6,506,282	2,817,112	2,124,806	1,564,364	6,515,375	6,503,433	0.18%

Table 6-9 Comparison of Volumes

The differences between the traditional method of calculating the composition of soil volumes and the summed volumes from the blocks results in an overall difference of comparable to the traditional methods.

6.5 PROTOTYPE VALIDATION: CASE STUDY

A case model is presented that allows us to validate the CAD-based simulation methodology prototype. For the purpose of this thesis, validation shall be defined as a comparison between the research simulation results and a method of analysis accepted by industry as being an accurate representation of the system. The case study shall compare a traditional method of estimating an earthmoving project, to the CAD-based simulation prototype, as a means of validation. The case study uses the excavation that has been used for display and verification throughout this Chapter. The case study is a 6,500,000 BCM excavation, containing a volume composition as described in Table 6-10. The excavation is divided into three benches, and the soil composition of each bench, is contained in Table 6-9.

Overburden (BCM)	Silt (BCM)	Clay (BCM)	Sand (BCM)
644,192	1,309,756	1,365,365	3,184,119

Table 6-10 Case Study: Excavation Soil Composition

The case study consists of a single source and a single placement, with one haul road connecting the two. The road between the source and the placement is described previously in Table 6-3. The loading method is a shovel with trucks loading on both sides and is described in Section 3.4.1 and Figures 3-11, 3-12 and 3-13.

The study case is analyzed using the CAD-based simulation model for earthmoving described in this Chapter. The simulation was run for 25 loads for each excavation block and weighted by volume to determine the results.

Simulation Input:

The excavation used the design and grid displayed in Figure 6-15. It was described using the Grid tool, and was defined in MicroStation by selecting the southwest and northeast extends of the excavation. The subgrids construction sequence was ordered as outlined in Appendix C. The loading operation used a shovel, loading trucks on both sides. The EX-3500 excavator was used for the entire excavation, and the neural network was used for productivity prediction. The road was defined using the Table 6-4. Six trucks were allocated to the process.

Other input parameters include the variables for the simulation high-level model. The global variables in the loading model were given values of XX(10) = 0.5 minutes, XX(11) = 0.5 minutes, XX(12) = 0.5 minutes, XX(14) = 4 trucks and XX(15) = 2 minutes. The global variable, intersection crossing time, was defined as XX(5) = 0.5 mintues.

All activity durations in the high-level model used a random linear distribution, distributed 10% above and 10% below the calculated values. The travel in placement loaded, dump and travel in placement empty activities were given an average value of six minutes, but were distributed using the linear distribution.

Simulation Output:

Table 6-11 illustrates the output parameters from the CAD-based simulation model. The predicted total project duration was 5410 hours. The average ideal productivity, as predicted by the neural networks was 1503 BCM/hr. The average cycle time was 29.90 minutes, and the trucks waited on average 0.47 minutes per cycle for the excavator.

Total Project Duration (hrs)	Avg. Loader Ideal Production (BCM / hr)	Avg. Actual Loader Production (BCM / hr)	Avg. Truck Loading Time (min)	Avg. Loader Utilization (%)	Avg. Truck Cycle Time (min)	Avg. Truck Wait Time (min/cycle)
5,410	1503	1201	4.03	87.5%	29.90	0.47

Table 6-11 Case Study: CAD-Based Simulation Output

6.5.2 Validation: Traditional Estimating Methods

Table 6-12 is a traditional means of estimating an excavation of this type. The table was created using the following steps:

1) First, an ideal production value for each type of soil structure is estimated based on historical records, experience and intuition. The productivity values (BCM/hr) used are representative of the average values of the productivities used as input for the testing and training of the neural networks.

2) The distances for the haul road (km) and the average distance in the source (km) are known from the site design. The next step is to estimate velocities to determine travel times. The velocities (km/hr) are based on average road conditions, data from the Caterpillar Handbook (1997) and experience. The travel-loaded and travel-empty durations (min) are calculated as the distance (km) divided by the velocity (km/min).

3) The capacity of the trucks is 100 BCM. The loading time (min) is equal to the truck capacity (BCM) divided by the ideal production (BCM/min).

4) The dump time (min) used in this estimate is constant at six minutes, as was used in the simulation model.

5) One minute positioning time is allocated to the trucks prior to the loading operation. This accounts for loader repositioning and time for the trucks to position.

6) The intersection activity, is a result of intersections on the haul road. The intersection is encountered once on the travel-loaded, and once on the travel-empty activities. Each encounter of the intersection is allocated 0.5 minutes duration, for a total of one minute.

7) The cycle time for the trucks (min) is the sum of the activity durations. The number of loads per hour is equal to 60 minutes per hour divided by the truck cycle time.

8) The excavator cycle time (min) is the sum of the loading time and the position time.

9) The excavator production (trucks/hr) is equal to 60 (min/hr) divided by the excavator cycle time (min). The production of the excavator (BCM/hr) is the number of trucks per hour multiplied by the capacity of the trucks (BCM).

10) The number of trucks required is the number of trucks required to maintain 100% loader utilization. It is equal to the truck cycle time (min) divided by the excavator cycle time (min). The number of trucks used is the number of trucks required rounded up.

11) The average wait time (min/cycle) is the difference between the number of trucks used and the number of trucks needed, multiplied by the excavator cycle time (min).

productivity (BCM/hr).

13) The total truck hours is the total project duration multiplied by the number of trucks.

Production		Overburden **1430**	Silt **1440**	Clay **1450**	Sand **1400**	
Haul Road Distance (km)		5.275	5.275	5.275	5.275	
Avg. Source Distance (km)		0.4	0.4	0.4	0.4	
Total Distance (km)		5.675	5.675	5.675	5.675	
Activity	**Speed**	**Duration**	**Duration**	**Duration**	**Duration**	
Load		4.20	4.17	4.14	4.29	
Haul, Loaded	35	9.73	9.73	9.73	9.73	
Intersections (1 both ways)		1.00	1.00	1.00	1.00	
Dump		6.00	6.00	6.00	6.00	
Return	40	8.51	8.51	8.51	8.51	
Position		1.00	1.00	1.00	1.00	
Truck Cycle Time		30.44	30.41	30.38	30.53	
Lds / Truck / Hr		**1.97**	**1.97**	**1.98**	**1.97**	
Truck Requirements						
LDS / Hr / Truck		1.97	1.97	1.98	1.97	
Excavator Cycle Time (min)		5.20	5.17	5.14	5.29	
Excavator Production (Trucks)		11.55	11.61	11.68	11.35	
Load Factor (BCM)		100.00	100.00	100.00	100.00	
Production (BCM / hr)		1154.78	1161.29	1167.79	1135.14	
Trucks Required		**5.86**	**5.89**	**5.91**	**5.78**	
Trucks Used		6.00	6.00	6.00	6.00	
Avg.Truck Wait Time per cycle (min) =		0.74	0.59	0.45	1.19	
Total Sand Volume (BCM) =		644,192	1,309,756	1,365,365	3,184,119	**6,503,432**
Total Project Duration (hrs) =		**557.8**	**1127.8**	**1169.2**	**2805.1**	**5,660**
Total Truck Hours =		3347.1	6767.1	7015.2	16830.3	**33,960**

Table 6-12 Case Study: Traditional Estimate

The predicted total project duration is 5660 hours. The average ideal productivity, weighted based on volume of different soil types is 1422 BCM/hr. The average weighted cycle time is 30.46 minutes, and the trucks wait on average 0.87 minutes per cycle for the excavator.

Table 6-13 compares the results between the CAD-based simulation, and a traditional estimate. The results indicate that the CAD-based system will produce results that are similar to the results that would be obtained using a traditional estimate. This was expected, as using different earthmoving analysis methods should not cause significant variation. The CAD-based system predicts that the total project duration will be 5410 hours, while the traditional estimate indicates that the total project duration will be 5660 hours, a difference of 250 hours, or 4.4%. The difference in the durations can be largely attributed the predicted ideal productivity of the CAD system being 5.7% greater than that of the traditional system.

	Total Project Duration (hrs)	Avg. Loader Ideal Production (BCM / hr)	Avg. Actual Loader Production (BCM / hr)	Avg. Truck Loading Time (min)	Avg. Loader Utilization (%)	Avg. Truck Cycle Time (min)	Avg. Truck Wait Time (min/cycle)
CAD-Based Simulation	5,410	1503	1201	4.03	87.5%	29.90	0.47
Traditional Estimating	5,660	1422	1148	4.22	90.4%	30.46	0.87
Difference (Absolute)	250	-81	-53	0.19	3%	0.56	0.40
Difference (%)	4.4%	-5.7%	-4.6%	4.5%	3.2%	1.8%	45.9%

Table 6-13 Case Study: Comparison Between Estimating Methods

Appendix C contains sample input and output for the simulation for the Case Study.

6.6 SAMPLE APPLICATION: SELECTION OF CONSTRUCTION METHODS

Many earthmoving construction projects permit the use of various methods of earthmoving, but usually they are restricted by logical project parameters such as project size, average cut and fill depths, and hauling distances. For example the question of using scrapers verse a backhoe and large trucks is seldom encountered. Issues such as likely to be encountered. This section uses the CAD-based earthmoving simulation system to analyze different construction methods for the earthmoving project described in the Case Study.

Three different construction methods have been identified. Each method is simulated twice, with different number of trucks used for each run.

1) The first construction method is the truck and shovel operation, loading both sides. The ideal production is predicted using the neural network. The ideal production of the shovel is determined from the neural networks. The excavation is divided into three benches. Two different simulation runs are executed, one using 6 trucks (method 1), and one using 7 trucks (method 2).
2) The second construction method involves using a truck and shovel operation, loading both sides. The ideal production is two thirds of the predicted productivity from the neural network. The excavation is divided into four benches. Two runs are executed, one using 4 trucks (method 3) and one using 5 trucks (method 4).
3) The third construction method involves using a backhoe and truck operation. The top loading drive-by method as described in Chapter 3 is used. The ideal production is two thirds of the predicted productivity from the neural network. The excavation is divided into four benches. Two runs are executed, one using 4 trucks (method 5), and one using 5 trucks (method 6).

The simulation used for this case study follows the basic cyclic simulation process. By changing the atomic loading models while maintaining the rest of the models constant, and using the site defined from the CAD models, the simulation is used known, the results from the simulation models will be indicative of the effectiveness of the loading operation and the interactions between the trucks and the loading units. The different construction methods are described below in Table 6-14.

Construction Method Number	1	2	3	4	5	6
Excavator Type	Shovel - 1	Shovel - 1	Shovel - 2	Shovel - 2	Backhoe - 3	Backhoe - 3
Excavator Avg. Production (BCM)	1500	1500	1000	1000	1000	1000
Number of Excavators	1	1	1	1	1	1
Number of Trucks (100 BCM)	6	7	4	5	4	5
Number of Benches	3	3	4	4	4	4
Average Bench Depth (m)	8	8	6	6	6	6

Table 6-14 Sample Application: Construction Methods

Construction Method Number	1	2	3	4	5	6
Total Project Duration (hrs)	5410	4848	8653	7380	8562	7590
Total Excavator Hours	5410	4848	8653	7380	8562	7590
Total Truck Hours	32460	33936	34612	36900	34248	37950
Avg. Loader Utilization	87.9%	100.0%	85.5%	100.0%	86.6%	100.0%
Truck Cycle Times (min)	29.9	31.3	31.9	34.0	31.6	35.0
Avg. Truck Waiting Time	0.47	1.87	0.23	2.37	0.31	3.75
Excavator 1 = $600 / hr	3,246,000	2,908,800				
Excavator 2,3 = $400 / hr			3,461,200	2,952,000	3,424,800	3,036,000
Truck = $200 / hr	6,492,000	6,787,200	6,922,400	7,380,000	6,849,600	7,590,000
Total Direct Costs ($)	9,738,000	9,696,000	10,383,600	10,332,000	10,274,400	10,626,000

Table 6-15 Sample Application: Simulation Output and Cost Analysis

The simulation runs 25 loads for each excavation block. The overall system results are determined from the weighted averages. Table 6-15 contains the results of the simulation from the prototype system.

Results of the simulation can be used in a cost analysis to determine which of the above construction methods will result in the lowest overall project cost. From the arbitrary equipment cost comparison, we see that the shovel and truck operation with the construction method, followed by the same operation using 6 trucks as the second most favorable method. Method 5, the backhoe top loading drive-by method using 4 trucks is found to be more effective than methods 3 and 4. Adding another truck to method 5 renders the system the least effective out of the 6 options. These arbitrary costs only consider the direct equipment costs. Including indirect costs in each method may produce different total costs.

6.7 MODEL LIMITATIONS

The developed prototype is limited specifically to a one source and one placement scenario for earthmoving. This is typical of mining and dam projects. It is not extendable to other types of earthmoving projects such as road building in its current state. The development of an entire CAD-based system is time consuming. For example, with the experience gained within, the development of a system for road building may take 200 hours.

The development of a special purpose CAD-based simulation system is a time consuming process, but since the system is reusable, if used often, significant time savings can be achieved. The process of designing an excavation and site, and then defining the system for simulation, using the prototype system takes less than two hours. Compare this to the time required to develop a system model of the earthmoving process in AweSim, which to develop and test would exceed 40 hours. To incorporate the amount of data considered in the CAD-based system would require an additional 100^+ hours. This may compare to 4 to 5 hours to perform a traditional estimate.

hours for minor changes, and only 20 to 30 minutes to rerun the previously defined system.

In summary, the development of CAD-based systems are time consuming and take considerably more effort to develop and analyze than using a traditional means of estimating.

6.8 CONCLUSIONS

This Chapter presented the development and testing of a prototype aimed at demonstrating the validity of the CAD-based simulation modeling methodology. The special purpose CAD-based earthmoving simulation system demonstrates that the high level of detail that the CAD system provides can be harnessed to improve simulation. The high level of detail used within the simulation models and neural networks provide a very detailed, dynamic approach to the entire earthmoving system, that traditional estimating methods do not provide (without performing exhaustive calculations).

The prototype also demonstrated that, by using modular modeling theory in a special purpose environment, the practitioner can be separated from the simulation environment, but still harnesses the power of simulation.

No conclusions are made on whether the prototype or traditional method is more accurate, as no study was performed to validate the method on actual historical projects. However, the system does provide the user with a high level of comfort because the site specifics have been incorporated into the analysis of the earthmoving process.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 CONCLUSIONS

This thesis presented the CAD-based simulation methodology for simulating construction projects from a CAD environment. The goal behind the CAD-based simulation was to use existing data structures (CAD drawings) to extract the simulation data, and define the simulation system, from predefined atomic models. This approach to modeling allows the user to take full advantage of the CAD drawing that is previously created by designers.

The primary objective of this research was to develop a methodology for CAD-based simulation for construction. This objective was realized in Chapter 4. The methodology for developing CAD-based simulation systems was proven through the development of the prototype in Chapter 6.

The secondary objective of this research was the development of a special purpose CAD-based simulation model for earthmoving, using modular simulation theory. This objective was realized through the development of the prototype in Chapter 6, which incorporated the work of Chapters 3, 4, 5 and 6. The earthmoving simulation atomic models focused on the loading operation. This work also proved additional benefits of the CAD-based simulation modeling methodology. It showed that once a special purpose CAD-based system is developed, that simulation using CAD design files can incorporate large amounts of site information into the simulation with minimal input from the user. As well the prototype proved that by incorporating modular simulation concepts into the tasks of high-level system model creation can be defined from CAD, using atomic models. The user is separated from the simulation environment, thus the user requires minimal simulation knowledge to run the prototype system. The task of data entry for the simulation is semi-automated from CAD. This reduces the time and effort required to define input to the simulation model.

The tertiary objective of the research was the use of neural networks for excavator productivity prediction. This objective was realized in Chapter 5. The neural network approach to predicting productivity of excavators in a field application, using site and soil conditions for input was successful.

The CAD-based simulation methodology backed up by the prototype proves that by using this methodology, the simulation can meet the objectives of a construction manager's ideal simulation system. The prototype requires minimal simulation knowledge. The prototype is easy to use, assuming that the user has knowledge of CAD. By incorporating the site specifics, and the immense amount of data, it also, above all else, provides a level of analysis beyond conventional techniques. No conclusions are made as to whether traditional methods of estimating and planning or the CAD-based prototype is more accurate, since no comparison to actual completed projects were performed.

7.2 RECOMMENDATIONS

Recommendations for improvements and further work, include, using more powerful interfacing methods, extending the work to AutoCAD, removing the simulation from AweSim, and improving the earthmoving models using optimization.

Simulation Based Recommendations

The AweSim simulation environment was used for demonstration purposes in this research. Removing the simulation from the AweSim environment can enhance the capabilities of this research. AweSim carries excess features that are not required for this type of work. The CAD-based work can be incorporated into a stand-alone system like the AP2_Earth special purpose earthmoving simulation system. This would allow more meaningful statistics output, in an environment intended solely for earthmoving simulation.

CAD-Based Recommendations

MicroStation BASIC macros were used for the development of the CAD extraction modules. The use of more powerful MicroStation interfacing methods, such as MicroStation Development Language will increase control of drawing and extracting data. This is a requirement for any work extending the CAD-based methodology.

Although MicroStation was used for this research, other CAD programs could be used. Extending this work to other CAD platforms such as AutoCAD can extend the research to a wider range of projects. The same theory described in Chapter 4, the CAD-based simulation methodology, can be used to extend this research into AutoCAD.

Earthmoving Recommendations

Future work relating to using modular components for the earthmoving process include the development of atomic models for scrapers, and loading models for wheeled loaders.

Another need to improve earthmoving simulation, is an optimization module. In particular, the number of trucks can be optimized. They should be monitored constantly, and the system should be able to add or remove trucks based on the results of the ongoing simulation.

REFERENCES

AbouRizk, S.M. and Shi, J. (1994). "Automated Construction-simulation optimization." *Journal of Construction Engineering and Management.,* ASCE, Vol. 120, No. 2, pp. 374-385.

AbouRizk, S.M. and Hajjar, D. (1998) "A Framework for Applying Simulation in Construction." Canadian Journal of Civil Engineering. Accepted for publication in November 1997.

AbouRizk, S.M. and Halpin, D.W. and Lutz, J.D. (1992). "State of the art in construction simulation." Proceedings of the 1992 Winter Simulation Conference, pp. 1271-1276.

AbouRizk, S.M. and Wales, R.J. (1997). "Combined discrete-event/continuous simulation for project planning." *Journal of Construction Engineering and Management*, ASCE, Vol. 123, No. 1, pp. 11-20.

Beliveau, Y. (1990. "Position Data integration for improved construction control." *Journal of Transportation Engineering*, ASCE, Vol. 116, No. 6, pp 836-843.

Beliveau, Y., Dal, .T and Dixit, S. (1993). "CAD applications for metrial handling in the construction environment." 5th International Conference of Computing in Civil Engineering, ASCE, pp125-128.

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

Brown, F., Cooper, G., Ford, S., Aouad, G., Brandon, P., Child, T., Kirkham, J., Oxman, R. and Young B. (1995). "Integrated Approach to CAD; modeling concepts in building design and construction." Design Studies, Vol. 16, No. 3, pp. 327-347.

Caterpillar Handbook (1997). Publication by Catepillar Inc., Peoria, Illinois, U.S.A.

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

Chao, Li-Chung and Skibniewski, M.J. (1994). "Estimating Construction Producivity: Neural-Network-Based Approach." *Journal of Computing in Civil Engineering*, ASCE, Vol. 8, No. 2, pp. 234 – 251.

Cherneff, J., Logcher, R. and Sriram, D. (1991). " Integrating CAD with construction-schedule generation." *Journal of Computing in Civil Engineering*, Vol. 5, No. 1, pp 64-84.

Needs." *Journal of Management in Engineering*, ASCE, Vol. 6, No. 4, pp. 394-415.

Euler, J. (1994). "Advantages of using 3-D CAD plant models on the construction site." *Microcomputers in Civil Engineering*, Vol. 9, No. 6, pp 435-444.

Halpin, D.W. (1973). An investigation of the use of simulation networks for modeling construction operations. Ph.D. thesis presented to the University of Illinois, at Urbanan Champaign, Illinois, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

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

Ibbs, C.W. (1987). "Future directions for computerized construction research." *Journal of Construction Engineering and Management*, ASCE, Vol. 112, No. 3, pp. 326-345.

Jawawardane, A.K.W., and Price, A.D.F. (1994). "A new approach for optimizing earth moving operations." Proceedings of the Instition of Civil Engineers, Transportation, Vol. 105, No. 4, pp.249 – 258.

Knowles, P (1997). "Estimating Construction Productivity using Neural Networks." Masters of Science Thesis, University of Alberta

Liu, L.Y. and Ioannou, P.G. (1992). "Graphical object-oriented discrete-event simulation system." Proceedings of the 1992 Winter Simulation Conference, 1285-1291.

Luna, J.J. (1992). "Hierarchical, modular concepts applied to an object-oriented simulation model development environment." Proceedings of the 1992 Winter Simulation Conference, 694-699.

McCabe, B., Saddi, W. and AbouRizk, S.M. (1996). "Using Neural Networks to Estimate Pipeline Productivity." University of Alberta Research Report.

Mahoney, J. and Tatum, C. (1994). "Construction site applications of CAD." *Journal of Construction Engineering and Management*, ASCE, Vol. 120, No. 3, pp. 617 – 631.

Manavazhi, M (1997). "A Configuration-Based Modeling Methodology for the Automated Generation of Simulation Models in Construction." Ph.D. thesis presented to the University of Alberta, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Martinez, J.C. and Ioannou, G.P (1994). "General purpose simulation with stroboscope." Proceedings of the 1994 Winter Simulation Conference, 1159-1166.

Construction." *Journal of Construction Engineering and Management*, ASCE, Vol. 117, No. 4, pp. 606-623.

Navon, R. (1995). "CAD/CAM based automatic production and construction-data extraction." Proceedings of the 1995 Construction Congress, ASCE, pp.347 – 354.

Norman, V. (1990). "AutoMod II." Proceedings of the 1990 Winter Simulation Conference, pp. 94 – 98.

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

Paulson, B.C., Jr., W.T., Koo, C.C. (1987). "Construction Operations Simulation by MicroComputer." *Journal of Construction Engineering and Management*, ASCE, Vol. 113, No. 2, pp. 302-314.

Pritsker, A.A.B., O'Reilly, J.J., and LaVal, D.K. (1997). *Simulation with Visual Slam and Awesim*, John Wiley & Sons, New York, 1997.

Portas, J. (1997). "Estimating Concrete Formwork Productivity" Masters of Science Thesis, University of Alberta

Shi, J. (1995). "Automated Modeling and Optimization for Construction Simulation." Ph.D. thesis presented to the University of Alberta, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Shi, J. and AbouRizk, S.M. (1994). "A resource_based simulation approach with application in earthmoving/strip mining." Proceedings of the 1994 Winter Simulation Conference, pp. 1124-1128.

Shi, J. and AbouRizk, S.M. (1997). " Resource-based modeling for construction simulation." *Journal of Construction Engineering and Management*, ASCE, Vol. 123, No. 1, pp. 26-33.

Skolnick, J.F. (1993). "A CAD-based construction simulation tool kit for construction planning." Proceeedings of the 5th International Conference on Computing in Civil and Building Engineering – V-ICCCBE, pp. 117 – 124.

Wales, R.J. (1994). "Incorporating weather effects in project simulation." Masters of Science Thesis, University of Alberta

Wichard, D., Bill, R., Gates, K.H., Yoshinaga, T. and Ohcoshi, S. (1989). "Construction CAE. Integration of CAD, simulation, planning and cost control." Proceedings of the American Power Conference, V. 51,Published by the Illinois Institute of Technology, Chicago, IL, pp. 983-987.

Zeigler, B.P. (1987). "Hierarchical, modular discrete-event modeling in an object-oriented environment," *Simulation*, Vol. 49, No. 5, pp. 219-230.

APPENDIX A

NEURAL NETWORKS FOR EXCAVATOR PRODUCTIVITY ESTIMATION

The raw data for the neural network training, testing, and raw data from the data collection are included within this Appendix. The neural network testing data is described by data type as "Test" and the network training data is described as "Train". For the purpose of this thesis, all productivity values have been collected from a general earthmoving contractor. In order to maintain confidentiality, all productivity data values have been factored by an undisclosed constant value. All productivity values were factored equally so that the full effect of their relationships can still be expressed in this thesis. All productivity values are not, and should not be construed as accurate representations of the production capabilities of the Hitachi EX-3500 shovel. Appendix A, contains the following work related to the neural networks used for excavator productivity estimation:

1) Training and testing data
2) Comparison between actual productivity and neural network predicted productivity for training data.
3) Comparison between actual productivity and neural network predicted productivity for testing data.
4) The recall procedure for the trained network.

Table A-1 Network Training and Testing Data

		Input													Output
		Soil Types (0 = no, 1 = yes)					Excavat or Footing	Soil Decription				Loader / Truck Positioning			
Data Type	Data Point	Sand	Silt	Clay	Rock	OB	Dry = 1, Wet = 0	Frozen (Yes=1 , No=0)	Soft (1) or Hard (0)	Swing Angle Degrees	Face Height (m)	Good Side (%)	Bad Side (%)	Both Sides (%)	Production (BCM/hr)
Test	1	0	0	1	0	1	1	0	0	90	4	0	0	100	1480
Test	2	0	0	0	1	0	1	0	0	90	5	0	0	100	1326
Test	3	0	0	0	0	1	1	0	1	87.5	6	0	0	100	1469
Test	4	0	0	1	0	1	1	0	0	90	7	0	0	100	1409
Test	5	0	0	1	0	1	1	0	0	90	6	0	0	100	1332
Test	6	1	0	1	0	1	1	0	0	90	6	100	0	0	1271
Test	7	0	0	1	0	1	1	0	0	90	8	0	0	100	1613
Test	8	0	0	1	0	1	1	0	0	90	6	0	0	100	1537
Test	9	0	0	1	0	1	1	0	1	90	2	0	0	100	1385
Test	10	0	0	0	0	1	1	0	1	87.5	6	0	0	100	1438
Test	11	0	0	1	0	1	1	0	0	90	6	0	0	100	1473
Test	12	0	0	0	1	0	0	0	1	90	7	0	0	100	1393
Test	13	1	0	1	0	1	1	0	1	90	5	0	0	100	1331
Test	14	0	0	1	0	1	1	0	0	90	3	0	0	100	1307
Test	15	0	0	1	0	1	0	0	1	90	9	0	0	100	1731
Test	16	0	0	0	0	1	0	0	1	85	6	0	0	100	1336
Test	17	0	0	1	0	1	1	0	0	90	7	0	0	100	1541
Test	18	0	0	0	1	0	0	0	0	90	5	20	20	60	1210
Test	19	0	0	0	0	1	0	0	1	90	6	0	0	100	1353
Test	20	1	0	0	0	1	0	0	0	90	5	0	0	100	1519
Test	21	0	0	0	0	1	0	0	1	85	5	0	0	100	1469
Test	22	0	0	0	0	1	0	0	1	85	8	0	0	100	1455
Test	23	0	0	0	0	1	0	0	1	85	7	0	0	100	1527
Test	24	0	0	0	0	1	0	0	1	85	5	0	0	100	1351
Test	25	1	0	1	0	1	1	0	0	90	5	0	0	100	1370
Test	26	1	0	0	0	1	0	0	1	90	5	0	0	100	1427
Train	27	0	0	0	0	1	0	0	1	90	6	0	0	100	1319
Train	28	0	0	0	0	1	1	0	1	90	6	0	0	100	1549
Train	29	0	0	1	0	1	1	0	0	90	6	0	0	100	1559
Train	30	0	0	0	0	1	0	0	1	92.5	6	0	0	100	1570
Train	31	0	0	0	0	1	1	0	1	85	4	33.3	33.3	33.3	1475
Train	32	0	0	1	0	0	1	1	0	90	5	100	0	0	1487
Train	33	0	0	0	0	1	0	0	1	85	5	50	0	50	1277
Train	34	0	0	1	0	1	1	0	0	90	7	0	0	100	1410
Train	35	0	0	1	0	1	0	1	0	90	6	10	10	80	1444
Train	36	0	0	1	0	1	0	1	0	90	6	0	0	100	1371
Train	37	0	0	1	0	1	1	1	0	90	6	25	25	50	1336
Train	38	0	0	0	0	1	0	1	1	90	5	50	50	0	1192
Train	39	1	1	1	0	1	0	1	1	80	6	33.3	33.3	33.3	1483
Train	40	0	0	0	0	1	0	1	1	90	6	33.3	33.3	33.3	1472
Train	41	0	0	0	0	1	1	1	0	90	5	50	50	0	1244
Train	42	0	0	0	0	1	0	1	1	85	6	100	0	0	1292
Train	43	0	0	0	0	1	0	1	1	75	5	50	0	50	1216

Train	44	0	0	1	0	1	1	1	0	90	5	0	0	100	1485
Train	45	0	0	1	0	1	1	0	0	90	4	60	0	40	1159
Train	46	0	0	1	0	1	0	1	0	90	5	20	0	80	1253
Train	47	0	0	0	1	0	0	0	0	90	7	0	0	100	1261
Train	48	0	0	0	1	0	0	0	0	90	3	0	0	100	1280
Train	49	1	0	1	0	1	1	0	1	90	3	0	0	100	1345
Train	50	0	0	0	0	1	1	0	1	87.5	4	0	0	100	1476
Train	51	0	0	1	0	1	0	0	1	90	5	0	0	100	1519
Train	52	0	0	0	0	1	1	0	0	90	5	0	0	100	1560
Train	53	0	0	0	0	1	0	0	1	90	6	0	0	100	1498
Train	54	0	0	1	0	1	1	0	0	90	7	0	0	100	1652
Train	55	0	0	1	0	1	1	0	0	90	8	0	0	100	1478
Train	56	0	0	1	0	1	1	0	0	90	6	0	0	100	1458
Train	57	0	0	1	0	1	1	0	0	90	10	0	0	100	1530
Train	58	0	0	1	0	1	1	0	0	90	9	0	0	100	1617
Train	59	0	0	0	0	1	0	0	1	90	7	0	0	100	1400
Train	60	0	0	0	0	1	0	0	1	90	7	0	0	100	1356
Train	61	0	0	1	0	1	1	0	0	90	9	0	0	100	1368
Train	62	0	0	1	0	1	1	0	0	90	6	0	0	100	1531
Train	63	0	0	1	0	1	1	0	0	90	6	0	0	100	1676
Train	64	0	0	1	0	1	1	0	0	90	8	0	0	100	1532
Train	65	0	0	1	0	1	1	0	0	90	10	0	0	100	1347
Train	66	0	0	1	0	1	1	0	0	90	7	0	0	100	1654
Train	67	0	0	0	0	0	1	0	0	87.5	6	0	0	100	1478
Train	68	0	0	0	0	1	0	0	1	87.5	8	0	0	100	1319
Train	69	0	0	1	0	1	1	0	0	90	9	0	0	100	1591
Train	70	0	0	0	1	0	0	0	0	87.5	5	0	0	100	1190
Train	71	1	1	1	0	1	0	0	1	90	5	0	0	100	1336
Train	72	0	0	1	0	1	1	0	0	90	6	0	0	100	1508
Train	73	0	0	0	0	1	0	0	1	87.5	8	0	0	100	1338
Train	74	0	0	0	0	1	0	0	1	87.5	7	0	0	100	1433
Train	75	0	0	1	0	1	1	0	0	90	10	0	0	100	1520
Train	76	0	0	1	0	1	1	0	0	90	10	0	0	100	1577
Train	77	0	0	0	0	1	0	0	1	90	7	0	0	100	1374
Train	78	0	0	1	0	1	1	0	0	90	6	0	0	100	1472
Train	79	0	0	1	0	1	1	0	0	90	10	0	0	100	1560
Train	80	0	0	0	1	1	0	0	0	87.5	6	0	0	100	1313
Train	81	0	0	1	0	1	1	0	0	90	3	0	0	100	1422
Train	82	0	0	0	0	1	0	0	1	87.5	3	25	25	50	1405
Train	83	0	0	1	0	1	1	0	0	90	6	0	0	100	1660
Train	84	0	0	0	0	1	0	0	1	87.5	5	25	25	50	1401
Train	85	0	0	1	0	1	1	0	0	90	4	0	0	100	1501
Train	86	1	1	1	0	1	0	1	0	90	3	10	10	80	1450
Train	87	0	0	0	1	1	0	0	0	85	6	0%	0%	100%	1373
Train	88	1	0	1	0	1	1	0	1	90	5	0%	0%	100%	1464
Train	89	1	1	1	0	1	1	0	1	90	3	25%	25%	50%	1494
Train	90	1	0	1	1	1	0	0	0	90	4	0%	0%	100%	1164
Train	91	0	0	0	1	1	0	0	0	85	6	0%	0%	100%	1142
Train	92	1	0	1	0	1	0	0	1	90	5	40%	60%	0%	1295
Train	93	0	0	0	0	1	1	0	1	90	7	0%	0%	100%	1630
Train	94	0	0	0	0	1	0	0	1	90	4	0%	0%	100%	1239
Train	95	0	0	0	0	1	1	0	1	90	5	0%	0%	100%	1382
Train	96	0	0	0	0	0	1	0	0	90	4	0%	0%	100%	1310
Train	97	0	0	1	0	1	1	0	0	90	5	0%	0%	100%	1555
Train	98	0	0	1	0	1	1	0	0	90	5	0%	0%	100%	1391
Train	99	0	0	1	0	1	1	0	0	90	7	0%	0%	100%	1383
Train	100	0	0	0	0	0	0	0	0	90	7	0%	0%	100%	1393
Train	101	0	0	1	0	1	1	0	0	90	7	0%	0%	100%	1293
Train	102	0	0	1	0	1	1	0	0	90	8	0%	0%	100%	1433

Train	104	0	0	1	0	1	0	0	0	90	9	40%	0%	60%	1409
Train	105	0	0	0	0	1	1	0	1	85	8	0%	0%	100%	1454
Train	106	1	0	1	0	1	1	0	0	90	4	0%	0%	100%	1509
Train	107	0	0	0	0	1	1	0	1	85	8	50%	0%	50%	1429
Train	108	0	0	0	0	1	1	0	0	90	7	0%	0%	100%	1502
Train	109	0	0	1	0	1	1	0	0	90	6	0%	0%	100%	1374
Train	110	0	0	0	0	1	1	0	1	90	6	0%	0%	100%	1585
Train	111	1	0	1	0	1	0	0	1	90	6	0%	0%	100%	1376
Train	112	0	0	0	0	1	1	0	1	85	5	50%	0%	50%	1380
Train	113	1	0	1	0	1	0	0	1	90	6	0%	0%	100%	1389
Train	114	0	0	0	0	1	1	0	1	85	5	50%	0%	50%	1313
Train	115	1	0	1	0	1	0	0	0	90	5	0%	0%	100%	1310
Train	116	0	0	0	0	1	1	0	1	85	7	0%	0%	100%	1444
Train	117	0	0	1	0	1	1	0	0	90	7	0%	0%	100%	1423
Train	118	0	0	1	1	0	1	0	1	85	4	50%	0%	50%	1254
Train	119	0	0	1	0	1	1	0	0	90	7	100%	0%	0%	1488
Train	120	1	0	1	0	1	1	0	1	90	7	10%	0%	90%	1517
Train	121	1	0	1	0	1	0	0	1	90	7	0%	0%	100%	1652
Train	122	1	0	1	0	1	0	0	1	90	4	0%	0%	100%	1386
Train	123	1	0	1	0	1	0	0	1	90	4	0%	0%	100%	1386
Train	124	1	0	0	0	1	1	0	0	90	4	0%	0%	100%	1298
Train	125	1	0	0	0	1	0	0	0	90	4	0%	0%	100%	1259
Train	126	0	0	0	0	1	0	0	1	85	5	0%	0%	100%	1329
Train	127	1	0	1	0	1	1	0	0	90	6	0%	0%	100%	1419
Train	128	0	0	0	0	1	0	0	1	85	5	0%	0%	100%	1319
Train	129	1	0	0	0	1	1	0	0	90	5	0%	0%	100%	1383
Train	130	0	0	0	0	1	1	0	1	85	8	0%	0%	100%	1684
Train	131	0	0	1	1	0	0	0	0	90	4	0%	0%	100%	1130
Train	132	1	0	1	0	1	1	0	1	90	9	20%	0%	80%	1644
										Average =					1421
										Standard Deviation =					125

2) Comparison between Actual Productivity and Neural Network Predicted Productivity for Training Data.

Table A-2 Actual verse Predicted Productivity: Training Data

Data Type	Data Point Number	Actual Productivity (BCM/hr)	Test- NN Productivity (BCM/hr)	Difference (BCM/hr)	Absolute Difference (BCM/hr)
Train	27	1319	1385	-66	66
Train	28	1549	1531	17	17
Train	29	1559	1497	62	62
Train	30	1570	1400	171	171
Train	31	1475	1397	78	78
Train	32	1487	1423	65	65
Train	33	1277	1326	-49	49
Train	34	1410	1505	-95	95
Train	35	1444	1370	74	74
Train	36	1371	1389	-18	18
Train	37	1336	1423	-87	87
Train	38	1192	1309	-117	117
Train	39	1483	1437	46	46
Train	40	1472	1376	96	96
Train	41	1244	1315	-71	71
Train	42	1292	1259	33	33

Train	44	1485	1433	52	52
Train	45	1159	1290	-131	131
Train	46	1253	1325	-72	72
Train	47	1261	1272	-11	11
Train	48	1280	1212	68	68
Train	49	1345	1381	-36	36
Train	50	1476	1412	64	64
Train	51	1519	1457	62	62
Train	52	1560	1444	116	116
Train	53	1498	1385	113	113
Train	54	1652	1505	146	146
Train	55	1478	1510	-32	32
Train	56	1458	1497	-39	39
Train	57	1530	1514	16	16
Train	58	1617	1513	104	104
Train	59	1400	1400	0	0
Train	60	1356	1400	-43	43
Train	61	1368	1513	-145	145
Train	62	1531	1497	33	33
Train	63	1676	1497	179	179
Train	64	1532	1510	22	22
Train	65	1347	1514	-167	167
Train	66	1654	1505	149	149
Train	67	1478	1450	28	28
Train	68	1319	1402	-84	84
Train	69	1591	1513	79	79
Train	70	1190	1215	-25	25
Train	71	1336	1440	-105	105
Train	72	1508	1497	11	11
Train	73	1338	1402	-64	64
Train	74	1433	1383	50	50
Train	75	1520	1514	6	6
Train	76	1577	1514	63	63
Train	77	1374	1319	55	55
Train	78	1472	1497	-25	25
Train	79	1560	1514	46	46
Train	80	1313	1283	30	30
Train	81	1422	1428	-6	6
Train	82	1405	1361	44	44
Train	83	1660	1497	163	163
Train	84	1401	1384	17	17
Train	85	1501	1462	38	38
Train	86	1450	1453	-3	3
Train	87	1373	1268	104	104
Train	88	1464	1505	-41	41
Train	89	1494	1465	29	29
Train	90	1164	1167	-3	3
Train	91	1142	1268	-126	126
Train	92	1295	1310	-15	15
Train	93	1630	1594	37	37
Train	94	1239	1369	-130	130
Train	95	1382	1477	-95	95
Train	96	1310	1346	-36	36
Train	97	1555	1484	71	71
Train	98	1391	1484	-93	93
Train	99	1383	1505	-122	122
Train	100	1393	1407	-13	13
Train	101	1293	1505	-212	212
Train	102	1433	1510	-77	77
Train	103	1562	1587	-25	25
Train	104	1409	1448	-39	39
Train	105	1454	1573	-119	119
Train	106	1509	1420	89	89
Train	107	1429	1466	-37	37

Train	109	1374	1497	-123	123
Train	110	1585	1531	54	54
Train	111	1376	1451	-75	75
Train	112	1380	1375	5	5
Train	113	1389	1451	-62	62
Train	114	1313	1375	-62	62
Train	115	1310	1325	-15	15
Train	116	1444	1509	-64	64
Train	117	1423	1505	-82	82
Train	118	1254	1243	11	11
Train	119	1488	1353	135	135
Train	120	1517	1551	-35	35
Train	121	1652	1522	130	130
Train	122	1386	1347	39	39
Train	123	1386	1347	39	39
Train	124	1298	1308	-10	10
Train	125	1259	1277	-18	18
Train	126	1329	1354	-25	25
Train	127	1419	1472	-53	53
Train	128	1319	1354	-35	35
Train	129	1383	1366	16	16
Train	130	1684	1573	111	111
Train	131	1130	1148	-18	18
Train	132	1644	1568	76	76
	Average	**1420**	**1421**		**64**

Min	**1130**
Max	**1684**
Range	**554**

Average Difference Absolute (%)	**Average Difference Range (%)**
4.5%	11.5%

3) Comparison between Actual Productivity and Neural Network Predicted Productivity for Testing Data.

Table A-3 Actual verse Predicted Productivity: Testing Data

Data Type	Data Point Number	Actual Productivity (BCM/hr)	Test- NN Productivity (BCM/hr)	Difference (BCM/hr)	Absolute Difference (BCM/hr)
Test	1	1480	1462	18	18
Test	2	1326	1248	78	78
Test	3	1469	1491	-23	23
Test	4	1409	1505	-96	96
Test	5	1332	1497	-165	165
Test	6	1271	1231	40	40
Test	7	1613	1510	103	103
Test	8	1537	1497	40	40
Test	9	1385	1384	1	1
Test	10	1473	1497	-24	24
Test	11	1393	1377	16	16
Test	12	1331	1505	-174	174
Test	13	1331	1505	-174	174
Test	14	1307	1428	-120	120
Test	15	1731	1670	61	61

Test	17	1541	1505	36	36
Test	18	1210	1318	-108	108
Test	19	1353	1385	-32	32
Test	20	1519	1295	224	224
Test	21	1469	1354	115	115
Test	22	1455	1384	71	71
Test	23	1527	1370	158	158
Test	24	1351	1354	-3	3
Test	25	1370	1452	-82	82
Test	26	1427	1338	89	89
	Average	**1421**	**1420**		**80**

Min	**1210**
Max	**1731**
Range	**521**

Average Difference Absolute (%)	**Average Difference Range (%)**
5.6%	15.3%

4) Recall Procedure for the Trained Network.

```
/* Mon Feb 02 19:25:25 1998 (recall.c) */
/* Recall-Only Run-time for <untitled> */
/* Control Strategy is: <backprop> */

#if __STDC__
#define  ARGS(x) x
#else
#define  ARGS(x) ()
#endif /* __STDC__ */

/* --- External Routines --- */
extern double tanh  ARGS((double));
/* *** MAKE SURE TO LINK IN YOUR COMPILER's MATH LIBRARIES *** */

#if __STDC__
int NN_Recall( void *NetPtr, float Yin[13], float Yout[1] )
#else
int NN_Recall( NetPtr, Yin, Yout )
void  *NetPtr; /* Network Pointer (not used) */
float Yin[13], Yout[1];  /* Data */
#endif /* __STDC__ */
{
    float  Xout[20], Xsum[20];  /* work arrays */
    long   ICmpT;  /* temp for comparisons */
    /* *** WARNING: Code generated assuming Recall = 0 *** */
    /* Read and scale input into network */
    Xout[2] = Yin[0] * (2) + (-1);
    Xout[3] = Yin[1] * (2) + (-1);
    Xout[4] = Yin[2] * (2) + (-1);
    Xout[5] = Yin[3] * (2) + (-1);
    Xout[6] = Yin[4] * (2) + (-1);
    Xout[7] = Yin[5] * (2) + (-1);
    Xout[8] = Yin[6] * (2) + (-1);
    Xout[9] = Yin[7] * (2) + (-1);
    Xout[10] = Yin[8] * (0.11428571) + (-9.5714286);
    Xout[11] = Yin[9] * (0.25) + (-1.5);
    Xout[12] = Yin[10] * (2) + (-1);
    Xout[13] = Yin[11] * (3.3333332) + (-1);
    Xout[14] = Yin[12] * (2) + (-1);
```

```
    /* Generating code for PE 0 in layer 3 */
    Xsum[15]  = (float)(0.21699533) + (float)(0.022080634) * Xout[2] +
        (float)(-0.68329227) * Xout[3] + (float)(0.057167489) * Xout[4] +
        (float)(0.13829257) * Xout[5] + (float)(-0.54246074) * Xout[6] +
        (float)(-0.28454658) * Xout[7] + (float)(0.52677196) * Xout[8] +
        (float)(-0.51114076) * Xout[9] + (float)(-0.3315976) * Xout[10] +
        (float)(-0.69696647) * Xout[11];
    Xsum[15] += (float)(0.73716235) * Xout[12] +
        (float)(-0.26604447) * Xout[13] + (float)(-0.55346471) * Xout[14];
    /* Generating code for PE 1 in layer 3 */
    Xsum[16]  = (float)(-0.4506292) + (float)(-0.14203532) * Xout[2] +
        (float)(0.071925908) * Xout[3] + (float)(0.80271095) * Xout[4] +
        (float)(-0.37870604) * Xout[5] + (float)(-0.14547946) * Xout[6] +
        (float)(0.6331951) * Xout[7] + (float)(0.53816211) * Xout[8] +
        (float)(-0.33680466) * Xout[9] + (float)(0.53705019) * Xout[10] +
        (float)(1.233441) * Xout[11];
    Xsum[16] += (float)(-0.14170808) * Xout[12] +
        (float)(0.20005959) * Xout[13] + (float)(0.48522183) * Xout[14];
    /* Generating code for PE 2 in layer 3 */
    Xsum[17]  = (float)(-0.19815284) + (float)(-0.23538175) * Xout[2] +
        (float)(-0.17489892) * Xout[3] + (float)(-0.23945943) * Xout[4] +
        (float)(-0.075971067) * Xout[5] + (float)(-0.27070451) * Xout[6] +
        (float)(0.13491476) * Xout[7] + (float)(0.10925089) * Xout[8] +
        (float)(0.14783055) * Xout[9] + (float)(0.31495845) * Xout[10] +
        (float)(-0.018839126) * Xout[11];
    Xsum[17] += (float)(0.19156657) * Xout[12] +
        (float)(0.095773198) * Xout[13] + (float)(-0.33583844) * Xout[14];
    /* Generating code for PE 3 in layer 3 */
    Xsum[18]  = (float)(0.12073801) + (float)(0.081306167) * Xout[2] +
        (float)(0.54453361) * Xout[3] + (float)(-0.94578457) * Xout[4] +
        (float)(0.063160844) * Xout[5] + (float)(0.19931476) * Xout[6] +
        (float)(-0.74456632) * Xout[7] + (float)(-0.36958015) * Xout[8] +
        (float)(0.70369482) * Xout[9] + (float)(0.25085509) * Xout[10] +
        (float)(-0.14545786) * Xout[11];
    Xsum[18] += (float)(-0.31535286) * Xout[12] +
        (float)(-0.036940955) * Xout[13] + (float)(-0.049479801) * Xout[14];
    /* Generating code for PE 0 in layer 3 */
    Xout[15] = tanh( Xsum[15] );
    /* Generating code for PE 1 in layer 3 */
    Xout[16] = tanh( Xsum[16] );
    /* Generating code for PE 2 in layer 3 */
    Xout[17] = tanh( Xsum[17] );
    /* Generating code for PE 3 in layer 3 */
    Xout[18] = tanh( Xsum[18] );
    /* Generating code for PE 0 in layer 4 */
    Xsum[19]  = (float)(-0.1457238) + (float)(-0.27338368) * Xout[15] +
        (float)(0.64315611) * Xout[16] + (float)(0.24077791) * Xout[17] +
        (float)(0.40775588) * Xout[18];
    Xout[19] = tanh( Xsum[19] );
    /* De-scale and write output from network */
    Yout[0] = Xout[19] * (381.73751) + (1435.4);

    /* Generating code for PE 0 in layer 4 */
    return( 0 );
}
```

MICROSTATION EARTHMOVING DATA EXTRACTION MODULES

The earthmoving CAD-based simulation system utilizes CAD extraction modules to define the site, the construction process and the types of simulation models to describe the system. Nine extraction modules are used in the earthmoving prototype. The macros that run these modules are, and the modules are:

1) Initialize: Init. tool
2) Grid: Define Source tool
3) PlaceGrid: Draw Grid tool
4) Process: Const. Order tool
5) SimulationType: Sim. Type tool
6) Excavator: Ex. Type tool
7) Volumes: Vol. Calc. tool
8) HaulRoads: Haul Roads tool
9) SourceRoad: Source Roads tool

These macros are described below.

1) INITIALIZE - MACRO TO INITIALIZE THE SITE

```
'***************************************************************
' This macro initializes the number of sources and the number
' of placements areas that will be in the model
'***************************************************************
Sub Main
Dim FileLocation as string, FileName as string
Open "C:\USEPWIN\PROGRAM\SimulationFile.txt" For Input as #1
Input #1, FileLocation
Close #1
FileName = FileLocation + "SiteDesc.txt"
Open FileName for output access write lock write as #1
   Dim NumSources as string, NumPlacements as string, i as integer, j as integer, Benches as string
```

```
'*****************************************************************
' FIRST DETERMINE THE NUMBER OF SOURCES IN THE SITE DESIGN
'*****************************************************************
  NumSources = MbeInputBox("Enter the number of sources","","")
  NumPlacements = MbeInputBox("Enter the number of placements","","")
  Outputtext = trim$(NumSources) + "," + "Number of Sources"
  Print #1, Outputtext
  Outputtext = trim$(NumPlacements) + "," + "Number of Placements"
  Print #1, Outputtext
  Sources = val(NumSources)
'*****************************************************************
' NOW DETERMINE THE NUMBER ORIGINAL GROUND SURFACE IN THE DESIGN
'*****************************************************************
  Intsurface = MbeInputBox("Enter the SURFACE NUMBER of the Original Ground","","")
  Message = "Enter the SURFACE NAME of original ground surface "
  intsurface2 = MbeInputBox(Message,"","")
  Message = " Original ground surface for the site"
  OutputText = trim$(Intsurface) + "," + trim$(intsurface2) + "," + Message
  Print #1, OutputText
'*****************************************************************
' NOW DETERMINE THE NUMBER OF SUBGRADE SURFACES (IE SOIL SURFACES)
'*****************************************************************
  Print #1, "(Define the number and designation of subgrade surfaces in the design)"
  Message = "Enter the NUMBER OF SUBSURFACE SURFACES, IE SOILS SURFACES within the design, "
  Message = message + "do not include the Original ground"
  Intsurface = MbeInputBox(Message,"","")
  OutputText = trim$(intsurface) + "," + "The number of subgrade surfaces"
  Print #1, OutputText
  Surfaces = val(intsurface)
  For i = 1 to Surfaces
    Message = "Enter the SURFACE NUMBER of subgrade surface(" + str$(i) + ")"
    Intsurface = MbeInputBox(Message,"","")
    Message = "Enter the SURFACE NAME of subgrade surface(" + str$(i) + ")"
    intsurface2 = MbeInputBox(Message,"","")
    OutputText = trim$(Intsurface) + "," + trim$(intsurface2)
    Print #1, OutputText
  Next
'*****************************************************************
'NOW DESCRIBE EACH SOURCE IN INDIVIDUAL SOURCE FILES
'*****************************************************************
For i = 1 to Sources
    Filename = FileLocation + "SourceDesc" + trim$(str$(i)) + ".txt"
    Open Filename for output access write lock write as #2
    Message = "Enter the number of Benches for Source(" + str$(i) + ")"
    Benches = MbeInputBox(Message,"","")
    Outputtext = trim$(str$(i)) + "," + "Source#"
    Print #2, Outputtext
    Outputtext = trim$(Benches) + "," + "Number of Benches"
    Print #2, Outputtext
'*****************************************************************
' NOW DETERMINE THE FINAL DESIGN SURFACE FOR EACH SOURCE
    Message = "Enter the final design SURFACE NUMBER for Source(" + str$(i) + ")"
    intsurface2 = MbeInputBox(Message,"","")
    Message = "Enter the final design SURFACE NAME for Source(" + str$(i) + ")"
    surfacename = MbeInputBox(Message,"","")
    Message = "FINAL design surface for source(" + trim$(str$(i)) + ")"
    OutputText = trim$(intsurface2) + ","+ trim$(surfacename) + "," + message
    Print #2, OutputText
    FGsurfacedesc = surfacename
    FGsurfacenum  = val(intsurface2)
'*****************************************************************
' NOW DETERMINE THE SURFACE FOR EACH BENCH IN EACH SOURCE
    NumBenches = val(benches) - 1
    Print #2, "(Definition of the bench surfaces)"
    Message = "Please define the surfaces that represent each bench!"
    Message = Message + " The program automatically considers the Final Design"
    Message = Message + " Grade a Bench and Need not be defined for the F.G.."
    For j = 1 to Numbenches
```

```
      Message = "Enter the SURFACE NAME for bench (" + str$(j) + ")"
      Intsurface2 = MbeInputBox(Message,"","")
      OutputText = trim$(str$(j)) + "," + trim$(intsurface) + "," + trim$(intsurface2)
      Print #2, OutputText
    Next
    Numbenches =numbenches + 1
    Print #2, str$(Numbenches) + "," + str$(FGsurfacenum) + "," + FGsurfacedesc
  Close #2
Next
Close #1
End Sub
```

2) GRID - MACRO TO DEFINE GRID AND WRITE CENTERPOINTS AND STAKES TO FILE

```
'******************************************************************
Sub main
Dim Xi as string, Yi as string, zi as string , x as integer, y as integer, z as integer
Dim startPoint As MbePoint, Origin as MbePoint, Midpoint as MbePoint, ActiveLvl as Integer, view%, status%
Dim point As MbePoint, point2 As MbePoint, NE as MbePoint, SW as MbePoint
Dim i as integer, j as integer, Midpointx as double, Midpointy as double, Midpointz as double
Dim Filename as string, FileLocation as string, NumI as integer, NumJ as integer, Area as integer, Wide as single
Dim Interval as integer, GridNum as string, GridInput as string, GridSize as Integer, SourceNum as Integer, Message as String
'******************************************************************
' THIS PORTION OF CODE DETERMINES THE ACTIVE SOURCE
'******************************************************************
  Message = MbeInputBox ("Enter the source number for the grid:", "1", "")
  SourceNum=Val(Message)
  Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
  Input #1, FileLocation
  Close #1
  Filename = Filelocation +  "Grid_" + trim$(str$(SourceNum)) + ".txt"
  Open Filename for Output as #1
'******************************************************************
' THIS PORTION OF CODE DETERMINES THE EXTEND OF THE GRID LINES
'******************************************************************
'   Display messages
    MbeWriteCommand "Place South West Extent of Source"
    MbeWritePrompt  "Enter S.W. Point"
'   Wait for a data point or a reset
    MbeGetInput MBE_DataPointInput, MBE_ResetInput
    If MbeState.inputType = MBE_ResetInput Then
      MbeMessageBox "Grid Contents Not Saved"
      MbeWriteCommand ""
      MbeWritePrompt  ""
      Exit Sub
    End If
    MbeMessageBox "South West Point Confirmed"
    status = MbeState.getInputDataPoint (point, view)
'THIS CODE ROUNDS THE POINTS TO THE NEAREST 10
    Midpointx = point.x
    Midpointy = point.y
    Midpointz = point.z
    Xi = format$(Midpointx,"###")
    Yi = format$(Midpointy,"###")
    Zi = format$(MidPointz,"###")
    Midpointx = Val(Xi) / 10
    Midpointy = Val(Yi) / 10
    Midpointz = Val(Zi) / 10
    Yi = format$(Midpointy,"###")
    Xi = format$(Midpointx,"###")
    Zi = format$(Midpointz,"###")
    Midpointx = Val(Xi) * 10
    Midpointy = Val(Yi) * 10
    Midpointz = Val(zi) * 10
    SW.x = Midpointx
    SW.y = Midpointy
```

```
Display messages
   MbeWriteCommand "Place North East Extent of Source"
   MbeWritePrompt "Enter N.E. Point"
'  Wait for a data point or a reset
   MbeGetInput MBE_DataPointInput, MBE_ResetInput
   If MbeState.inputType = MBE_ResetInput Then
      MbeMessageBox "Grid Contents Not Saved"
      MbeWriteCommand ""
      MbeWritePrompt ""
      Exit Sub
   End If
   MbeMessageBox "NorthEast Point Confirmed"
   status = MbeState.getInputDataPoint (point2, view)
   MbeWritePrompt ""
'THIS CODE ROUNDS THE POINTS TO THE NEAREST 10
   Midpointx = point2.x
   Midpointy = point2.y
   Midpointz = point2.z
   Xi = format$(Midpointx,"###")
   Yi = format$(Midpointy,"###")
   Zi = format$(MidPointz,"###")
   Midpointx = Val(Xi) / 10
   Midpointy = Val(Yi) / 10
   Midpointz = Val(Zi) / 10
   Yi = format$(Midpointy,"###")
   Xi = format$(Midpointx,"###")
   Zi = format$(Midpointz,"###")
   Midpointx = Val(Xi) * 10
   Midpointy = Val(Yi) * 10
   Midpointz = Val(zi) * 10
   NE.x = Midpointx
   NE.y = Midpointy
   NE.z = Midpointz
Message = "North East Extent of the Grid:"
MbeMessagebox Message + Chr$(10) + str$(Ne.x) + chr$(10) +str$(Ne.y) + chr$(10)+str$(Ne.z)
Message = "South West Extent of the Grid:"
MbeMessagebox Message + Chr$(10)+ str$(SW.x) + chr$(10) +str$(sw.y) + chr$(10)+str$(sw.z)
Message = MbeInputBox ("Enter a Grid Size:", "50", "")
  Gridsize=Val(Message)
  NumI = (NE.x - SW.x)/GridSize
  NumJ = (NE.y - SW.y)/GridSize
  Print #1, trim$(str$(Gridsize))
'*******************************************************************
' THIS PORTION OF THE CODE WRITES THE GRID INFORMATION TO FILE
'*******************************************************************
For i = 1 to NumI
  For j = 1 to NumJ
    startPoint.x = SW.x + (GridSize* (i-1))
    startPoint.y = SW.y + (GridSize*(j-1))
    startPoint.z = SW.z
    MidPoint.x = startpoint.x + (GridSize/2)
    MidPoint.y = startpoint.y + (GridSize /2)
    Midpoint.z = startpoint.z
    Midpointx = midpoint.x
    Midpointy = midpoint.y
    Midpointz = midpoint.z
    Xi = format$(Midpointx,"###.000")
    Yi = format$(Midpointy,"###.000")
    Zi = format$(MidPointz,"###.000")
' Print the SubGrids information to file
    Area =GridSize^2
    Ar$ = format$(Area,"###")
    GridNum = trim$(str$(i)) + "," + trim$(str$(j))
    Print #1, gridNum + "," + trim$(xi) + "," + trim$(yi) + "," + trim$(zi) + "," + trim$(ar)
    MbeSendReset
  Next
Next
Close #1
Staking NumI, NumJ,Gridsize, SourceNum
```

```
' This subroutine stakes the grid at the centerpoints
'*******************************************************************
Sub Staking (NumI as integer, NumJ as Integer,Gridsize as Integer, SourceNum as integer)
   Dim startPoint As MbePoint, point As MbePoint, point2 As MbePoint, num as integer, OG as integer
   Dim OGstring as string, FGstring as string, FGing as integer, PointInt as integer, PointNum as string
   Dim Xi as string, Yi as string, zi as string, x as double, y as double, z as double, numsurfaces as integer, Filename as string
   Dim FileLocation as string, Message as String, ActiveLvl as Integer
'*******************************************************************
' First determine the number of surfaces and which one is the OG
   Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
   Input #1, FileLocation
   Close #1
   Filename = Filelocation + "SiteDesc.txt"
   Open Filename for Input as #1
   Input #1, num,message
   Input #1, num,message
   Input #1, num,OG, message
   OG = OG - 1
   Input #1, message
   Input #1, num, message
   Numsurfaces = num
   Close #1
   Filename = Filelocation + "SourceDesc" + trim$(str$(SourceNum)) + ".txt"
   Open Filename for Input as #1
   Input #1, num, message
   Input #1, num, message
   Numsurfaces = num + Numsurfaces
   Close #1
   OGstring = trim$(str$(OG)) + "&"
   OGString = trim$(OGstring)
'*******************************************************************
' Now set the Active level for the staking output
'  Set a variable associated with a dialog box
   ActiveLvl = 59 - SourceNum
   Message = "ACTIVE LEVEL " + TRIM$(Str$(ActiveLvl))
   MbeSendCommand Message
   MbeSendCommand "ACTIVE COLOR 17"
   Filename = Filelocation + "Grid_" + trim$(str$(SourceNum)) + ".txt"
   Num = numI * numJ
'*******************************************************************
' Now run a loop to stake all the centerpoints for each surface
For i = 1 to numsurfaces
   Open Filename for Input as #1
   FGing=i-1
   FGString = trim$(str$(FGing)) + "&"
   MbeSetAppVariable "SMMDCOMB", "sdtemp.point_num", 10000& + (i)*1000
   MbeSetAppVariable "SMMDCOMB", "sdtemp.original", 0&
   MbeSetAppVariable "SMMDCOMB", "sdtemp.final", 0& + i
   MbeSendCommand "STAKE SPOT "
   Input #1, message
   For j = 1 to num
      Input #1, gridx,gridy,xi,yi,zi,area
      x=val(Xi)
      y=val(yi)
      z=val(zi)
      startPoint.x = x
      startPoint.y = y
      startPoint.z = z
      Point.x = startpoint.x
      Point.y = startpoint.y
      Point.z = startpoint.z
      MbeSendDataPoint point, 1%
   Next
   MbeSendReset
   Close #1
Next
'*******************************************************************
' THIS AREA IS FOR EXPORTING THE DATA TO FILE
```

```
   MbeSetAppVariable "DTMDTRAN", "exportG.point_from", 10000&
   MbeSetAppVariable "DTMDTRAN", "exportG.point_to", 20000&
   Filename = FileLocation + "Stakes.txt"
   MbeSetAppVariable "DTMDTRAN", "exportG.export_file", Filename
End Sub
```

3) PLACEGRID - MACRO TO DRAW GRID IN CAD AND LABELS CENTERPOINTS

```
'*********************************************************************
' This macro places a grid on the source where the OG - FG stake results
' a cut stake.  It also places the grid descriptor text
'*********************************************************************
Sub main
Dim Xi as double, Yi as double, Zi as double, Gridx as integer, Gridy as integer, startPoint As MbePoint,
Dim Origin as MbePoint, Midpoint as MbePoint, Midpointx as double, Midpointy as double, Midpointz as double
Dim point As MbePoint, point2 As MbePoint, NE as MbePoint, SW as MbePoint, i as integer, j as integer
Dim Dim Filename as string, FileLocation as string, NumI as integer, NumJ as integer
Dim Area as integer, Wide as single, Interval as integer, GridNum as string, SourceNum as Integer, Message as String
Dim GridInput as string, GridSize as Integer, ActiveLvl as Integer,  view%, status%, Textsize as single, NumRecords as integer
'*********************************************************************
' THIS PORTION OF CODE DETERMINES THE ACTIVE SOURCE
'*********************************************************************
Message = MbeInputBox ("Enter the source number for the grid:", "1", "")
SourceNum=Val(Message)
Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
Input #1, FileLocation
Close #1
Filename = Filelocation +  "Grid_" + trim$(str$(SourceNum)) + ".txt"
Open Filename for Input as #1
Input #1, gridsize
Close #1
Textsize = Gridsize / 15
Filename = Filelocation +  "GridtoMS_" + trim$(str$(SourceNum)) + ".txt"
Open Filename for Input as #1
Input #1, Numrecords
'*********************************************************************
' THIS PORTION OF THE CODE PLACES A GRID AND CENTERPOINTS ON THE DRAWNG
'*********************************************************************
ActiveLvl = 59 - SourceNum
Message = "ACTIVE LEVEL " + TRIM$(Str$(ActiveLvl))
MbeSendCommand Message
MbeSendCommand "ACTIVE COLOR 7"
For i = 1 to NumRecords
   Input #1, Gridx,Gridy,Xi,Yi,Zi
   Message = trim$(Str$(Gridx)) + "," + trim$(str$(Gridy))
   MbeSendCommand "PLACE TEXT ICON "
   MbeSetScaledAppVar "", "tcb->chheight", textsize
   MbeSetScaledAppVar "", "tcb->chwidth", textsize
   MbeSendKeyin Message
   MbeSendAppMessage "TEXTEDIT",""
   Point.x = Xi
   Point.y = Yi
   Point.z = Zi
   MbeSendDataPoint point, 1%
   MbeSendReset
   MbeSendCommand "PLACE SMARTLINE "
   startPoint.x = Xi - (Gridsize / 2)
   startPoint.y = Yi - (gridsize / 2)
   startPoint.z = Zi
   point.x = startPoint.x
   point.y = startPoint.y
   point.z = startPoint.z
   MbeSendDataPoint point, 1%
   point.x = startPoint.x + Gridsize
   point.y = startPoint.y
   point.z = startPoint.z
```

```
    point.y = startPoint.y + Gridsize
    point.z = startPoint.z
    MbeSendDataPoint point, 1%
    point.x = startPoint.x
    point.y = startPoint.y + Gridsize
    point.z = startPoint.z
    MbeSendDataPoint point, 1%
    point.x = startPoint.x
    point.y = startPoint.y + Gridsize
    point.z = startPoint.z
    MbeSendDataPoint point, 1%
    point.x = startPoint.x
    point.y = startPoint.y
    point.z = startPoint.z
    MbeSendDataPoint point, 1%
    MbeSendReset
Next
MbeSendCommand "CHOOSE ELEMENT "
Close #1
End Sub
```

4) PROCESS - **MACRO TO DEFINE EXCAVATION PROCESS BY SELECTING LABELS IN THE DESIRED CONSTRUCTION ORDER**

```
'*****************************************************************
' Example of element location techniques
'-----------------------------------------------------------------
' Copyright (1995) Bentley Systems, Inc., All rights reserved.
' $Workfile: locate.bas $
' $Revision: 6.3 $
'    $Date: 01 Sep 1995 11:50:44 $
' "MicroStation" is a registered trademark of Bentley Systems, Inc.
' Limited permission is hereby granted to reproduce and modify this
' copyrighted material provided that the resulting code is used only
' in conjunction with Bentley Systems products under the terms of the
' license agreement provided therein, and that this notice is retained
' in its entirety in any such reproduction or modification.

Function locate_pointsCloseEnough (point1 as MbePoint, point2 as MbePoint) as Integer
  If Abs (point1.x - point2.x) + Abs (point1.y - point2.y) _
          < MbeState.LocateTolerance Then
    locate_pointsCloseEnough = TRUE
  End If
End Function

Sub locate_manipulateElement (point as MbePoint, view as Integer)
  Dim elem as New MbeElement, filePos as Long
  ' turn off graphics while changing element
  MbeState.noElementDisplay = 1
  ' start change color and relocate element
  MbeSendCommand "CHANGE COLOR"
  MbeRelocate point, view
  MbeSendDataPoint point, view
  ' start change weight and relocate element
  MbeSendCommand "CHANGE WEIGHT"
  MbeRelocate point, view
  MbeSendDataPoint point, view
  MbeSendCommand "LOCELE"
  MbeRelocate point, view
  MbeState.noElementDisplay = 0
  ' get the element to redraw it with its new symbology
  filePos = elem.fromLocate()
  elem.display MBE_NormalDraw
  ' set MicroStation to a neutral state.
  MbeSendCommand "NULL"
End Sub
```

```
'   locate_isElementAcceptable - returns TRUE if element passes
'               filter criteria and FALSE otherwise.
'---------------------------------------------------------------
Function locate_isElementAcceptable (elem as MbeElement) as Integer
  dim range as MbeRange, origin as MbePoint, endPnt as MbePoint, points() as MbePoint, cellLevels() as Integer
  dim cellBox() as MbePoint, iPoint as Integer, indent as Integer, ts$ as String, enterDataFlds() as MbeEDField
  dim numEDFields as Integer, iEDField as Integer
  If elem.type = MBE_Text Then
    locate_isElementAcceptable = TRUE
  End If
  If Elem.type = MBE_Text Then
    If Elem.getString (ts$, numEDFields, enterDataFlds) = MBE_Success Then
      print #1, ts$
      If numEDFields > 0 Then
        For iEDField = LBound (enterDataFlds) to UBound (enterDataFlds)
          'print #1, Spc(indent);"edf ";iEDField;
          'print #1, " start ";enterDataFlds(iEDField).start;
          'print #1, " length ";enterDataFlds(iEDField).length;
          'print #1, " just ";enterDataFlds(iEDField).justification
        Next iEDField
      End If
    End If
  End If
End Function

Sub locate_elementUnacceptable
  MbeWriteStatus "Only text allowed"
End Sub

Sub locate_displayPrompts
  MbeWriteCommand "Locate Test "
  MbeWritePrompt  "Identify line"
End Sub

Sub locate_selectElementsToModify(Sourcenum as integer)
  Dim elem as New MbeElement,  filePos as Long, point as MbePoint, acceptScreenPt as MbePoint,  firstScreenPt as MbePoint
  Dim stat as Integer, view as Integer, acceptPoint as MbePoint, acceptView as Integer, continueLocate as Integer
  Dim haveAcceptPoint as Integer Filename as String, Filelocation as string
  continueLocate     = FALSE
  haveAcceptPoint    = FALSE
  Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
  Input #1, FileLocation
  Close #1
  Filename = "ConstructionProcess_" + trim$(str$(SourceNum)) + ".txt"
  Filename = Filelocation + "\Excavate\" + Filename
  Open Filename for Output as #1
  Do
    Call locate_displayPrompts ()
    MbeStartLocate 0,0,0,continueLocate
    If haveAcceptPoint Or continueLocate Then
      Call MbeSendDataPoint (acceptPoint, acceptView)
      continueLocate = FALSE

      If haveAcceptPoint Then
        haveAcceptPoint = FALSE
        point = acceptPoint
        view  = acceptView
        firstScreenPt = acceptScreenPt
      End If
    Else
      ' Wait for a data point or a reset
      MbeGetInput MBE_DataPointInput, MBE_ResetInput
      ' If user resets, exit the loop.
      If MbeState.inputType = MBE_ResetInput Then
        Exit Do
      ' On a data point, retrieve the point for future use, then
      '  send it through to be processed.
      ElseIf MbeState.inputType = MBE_DataPointInput Then
        stat = MbeState.getInputDataPoint (point, view, firstScreenPt)
```

```
      End If
   End If

   ' If we found an element, check for acceptability
   If MbeState.cmdResult = MBE_AcceptQuery Then
      ' if we did not find a line, keep resetting until
      '   either we find one or we run out of candidates
      Do
         filePos = elem.fromLocate ()
         If locate_isElementAcceptable (elem) Then
            elem.display MBE_Hilite
            MbeWritePrompt "Accept / Reject"
            MbeGetInput MBE_DataPointInput, MBE_ResetInput
            If MbeState.inputType = MBE_DataPointInput Then
               ' on Data point, accept element and modify it
               stat = MbeState.getInputDataPoint (acceptPoint, _
                                acceptView, acceptScreenPt)
               Call locate_manipulateElement (point, view)
               continueLocate = locate_pointsCloseEnough (acceptScreenPt, _firstScreenPt)
               haveAcceptPoint = TRUE
            Else
               elem.display MBE_NormalDraw
               acceptPoint = point
               acceptView = view
               continueLocate = TRUE
            End If
            Exit Do
         Else
            MbeSendReset
            If MbeState.cmdResult = MBE_ElementNotFound Then
               Call locate_elementUnacceptable
               Exit Do
            End If
         End If
      Loop While elem.type <> MBE_Text
   ElseIf MbeState.cmdResult = MBE_ElementNotFound Then
      MbeWriteError "Element Not Found"
   Else
      print "unexpected MbeState.cmdResult of "; MbeState.cmdResult
   End If
  Loop
  Close #1
End Sub
'-----------------------------------------------------------------
' Main - main subroutine
Sub Main
   Dim saveColor as Integer, saveWeight as Integer, saveMsgs as Integer, elemSet as New MbeElementSet
   Dim message as string, SourceNum as integer
   ' set up for manipulation by saving the old settings and setting new ones
   saveColor        = MbeSettings.color
 ' set MicroStation to a neutral state
   MbeSendCommand "NULL"
   ' get rid of any selection set
   If elemSet.fromSelectionSet (1) = MBE_Success Then
      elemSet.clear
   End If
   MbeSettings.color  = 3
   MbeWriteStatus "Leaving locate function"
   Message = MbeInputBox ("Enter the source number for the grid:", "1", "")
   SourceNum=Val(Message)
   locate_selectElementsToModify sourcenum
   ' Reset settings to saved settings
   MbeState.messages  = saveMsgs
   MbeWritePrompt
   MbeWriteCommand
End Sub
```

BY SELECTING LABELS FOR THE DESIRED
EXCAVATION (SIMULATION) TYPE

```
'***********************************************************************
' Example of element location techniques
'-------------------------------------------------------------------
' Copyright (1995) Bentley Systems, Inc., All rights reserved.
'  $Workfile:  locate.bas  $
'  $Revision:  6.3  $
'     $Date:  01 Sep 1995 11:50:44  $
' "MicroStation" is a registered trademark of Bentley Systems, Inc.
' Limited permission is hereby granted to reproduce and modify this
' copyrighted material provided that the resulting code is used only
' in conjunction with Bentley Systems products under the terms of the
' license agreement provided therein, and that this notice is retained
' in its entirety in any such reproduction or modification.

Function locate_pointsCloseEnough (point1 as MbePoint, point2 as MbePoint) as Integer
   If Abs (point1.x - point2.x) + Abs (point1.y - point2.y) _
            < MbeState.LocateTolerance Then
      locate_pointsCloseEnough = TRUE
   End If
End Function

Sub locate_manipulateElement (point as MbePoint, view as Integer)
   Dim elem as New MbeElement, filePos as Long

   ' turn off graphics while changing element
   MbeState.noElementDisplay = 1

   ' start change color and relocate element
   MbeSendCommand "CHANGE COLOR"
   MbeRelocate point, view
   MbeSendDataPoint point, view

   ' start change weight and relocate element
   MbeSendCommand "CHANGE WEIGHT"
   MbeRelocate point, view
   MbeSendDataPoint point, view
   MbeSendCommand "LOCELE"
   MbeRelocate point, view
   MbeState.noElementDisplay = 0

   ' get the element to redraw it with its new symbology
   filePos = elem.fromLocate()
   elem.display MBE_NormalDraw

   ' set MicroStation to a neutral state.
   MbeSendCommand "NULL"

End Sub

Function locate_isElementAcceptable (elem as MbeElement,ExType as integer) as Integer
   dim range as MbeRange, origin as MbePoint, endPnt as MbePoint, cellBox() as MbePoint
   dim points() as MbePoint, cellLevels() as Integer, iPoint as Integer, indent as Integer
   dim enterDataFlds() as MbeEDField, numEDFields as Integer
   dim iEDField as Integer, ExString as String, ts$  as String

   ExString = trim$(str$(ExType)) + ","
   If elem.type = MBE_Text Then
      locate_isElementAcceptable = TRUE
   End If
   If Elem.type = MBE_Text Then
      If Elem.getString (ts$, numEDFields, enterDataFlds) = MBE_Success Then
         print #1, ExString, ts$
      End If
   End If
End Function
```

```
End Sub

Sub locate_displayPrompts
   MbeWriteCommand "Locate Test "
   MbeWritePrompt  "Identify line"
End Sub

Sub locate_selectElementsToModify(Sourcenum as integer, ExType as integer)
   Dim elem as New MbeElement, filePos as Long
   Dim point as MbePoint, acceptPoint as MbePoint, stat as Integer, view as Integer
   Dim acceptScreenPt as MbePoint,  firstScreenPt as MbePoint
   Dim acceptView as Integer, continueLocate as Integer
   Dim haveAcceptPoint as Integer, Filename as String, Filelocation as string

   continueLocate     = FALSE
   haveAcceptPoint   = FALSE
   Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
   Input #1, FileLocation
   Close #1
   Filename = "ExcavationType_" + trim$(str$(SourceNum)) + ".txt"
   Filename = Filelocation + "\Excavate\" + Filename
   Open Filename for Append access write as #1
   Do
      Call locate_displayPrompts ()
      MbeStartLocate 0,0,0,continueLocate
      If haveAcceptPoint Or continueLocate Then
         Call MbeSendDataPoint (acceptPoint, acceptView)
         continueLocate = FALSE

         If haveAcceptPoint Then
            haveAcceptPoint = FALSE
            point = acceptPoint
            view  = acceptView
            firstScreenPt = acceptScreenPt
         End If
      Else
         MbeGetInput MBE_DataPointInput, MBE_ResetInput
         If MbeState.inputType = MBE_ResetInput Then
            Exit Do
         ElseIf MbeState.inputType = MBE_DataPointInput Then
            stat = MbeState.getInputDataPoint (point, view, firstScreenPt)
            MbeSendLastInput
         End If
      End If
      If MbeState.cmdResult = MBE_AcceptQuery Then
         Do
            filePos = elem.fromLocate ()
            If locate_isElementAcceptable (elem,ExType) Then
               elem.display MBE_Hilite
               MbeWritePrompt "Accept / Reject"
               MbeGetInput MBE_DataPointInput, MBE_ResetInput
               If MbeState.inputType = MBE_DataPointInput Then
                  stat = MbeState.getInputDataPoint (acceptPoint, _
                                 acceptView, acceptScreenPt)
                  Call locate_manipulateElement (point, view)
                  continueLocate = locate_pointsCloseEnough (acceptScreenPt, _firstScreenPt)
                  haveAcceptPoint = TRUE
               Else
                  elem.display MBE_NormalDraw
                  acceptPoint = point
                  acceptView = view
                  continueLocate = TRUE
               End If
               Exit Do
            Else
               MbeSendReset
               If MbeState.cmdResult = MBE_ElementNotFound Then
                  Call locate_elementUnacceptable
```

```
      End If
    Loop While elem.type <> MBE_Text
  ElseIf MbeState.cmdResult = MBE_ElementNotFound Then
    MbeWriteError "Element Not Found"
  Else
    print "unexpected MbeState.cmdResult of "; MbeState.cmdResult
  End If
 Loop
 Close #1
End Sub

Sub Main
 Dim elemSet as New MbeElementSet, message as string, SourceNum as integer, descinput as string
 Dim ExType as integer, Filename as string, FileLocation as string, exnum as integer, i as integer, Ex() as string   ' set up for
 Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
 Input #1, FileLocation
 Close #1
 Filename = Filelocation + "\Excavate\" + "ExType.txt"
 Open Filename for Input as #1
 Line Input #1, descinput
   ExNum = val(descinput)
 Close #1
'CREATE THE ARRAY FOR EXCAVATION TYPES
 Redim ex(exnum)
 For i = 1 to ExNum
   Ex(i) = "Excavation Type (" + trim$(str$(i)) + ")"
 Next
' set MicroStation to a neutral state
 MbeSendCommand "NULL"
 ' get rid of any selection set
 If elemSet.fromSelectionSet (1) = MBE_Success Then
   elemSet.clear
 End If
 MbeSettings.color  = 5
 MbeWriteStatus "Leaving locate function"
 Message = MbeInputBox ("Enter the source number for the grid:", "1", "")
 SourceNum=Val(Message)
 ExType% = MbeSelectbox("Select Excavation Type",Ex(),"")
 locate_selectElementsToModify sourcenum,ExType
 MbeWritePrompt
 MbeWriteCommand
End Sub
```

6) EXCAVATOR - MACRO TO DEFINE THE EXCAVATOR TYPE BY SELECTING LABELS

```
'*******************************************************************
' Example of element location techniques
'-------------------------------------------------------------------
' Copyright (1995) Bentley Systems, Inc., All rights reserved.
' $Workfile: locate.bas $
' $Revision: 6.3 $
'   $Date: 01 Sep 1995 11:50:44 $
' "MicroStation" is a registered trademark of Bentley Systems, Inc.
' Limited permission is hereby granted to reproduce and modify this
' copyrighted material provided that the resulting code is used only
' in conjunction with Bentley Systems products under the terms of the
' license agreement provided therein, and that this notice is retained
' in its entirety in any such reproduction or modification.

Function locate_pointsCloseEnough (point1 as MbePoint, point2 as MbePoint) as Integer
  If Abs (point1.x - point2.x) + Abs (point1.y - point2.y) _
         < MbeState.LocateTolerance Then
    locate_pointsCloseEnough = TRUE
  End If
End Function
```

```
' turn off graphics while changing element
   MbeState.noElementDisplay = 1
' start change color and relocate element
   MbeSendCommand "CHANGE COLOR"
   MbeRelocate point, view
   MbeSendDataPoint point, view
' start change weight and relocate element
   MbeSendCommand "CHANGE WEIGHT"
   MbeRelocate point, view
   MbeSendDataPoint point, view
   MbeSendCommand "LOCELE"
   MbeRelocate point, view
   MbeState.noElementDisplay = 0
' get the element to redraw it with its new symbology
   filePos = elem.fromLocate()
   elem.display MBE_NormalDraw
' set MicroStation to a neutral state.
   MbeSendCommand "NULL"
End Sub

Function locate_isElementAcceptable (elem as MbeElement,ExType as integer) as Integer
   dim range as MbeRange, origin as MbePoint, endPnt as MbePoint, cellBox() as MbePoint
   dim points() as MbePoint, cellLevels() as Integer, iPoint as Integer, indent as Integer
   dim ts$ as String, enterDataFlds() as MbeEDField, numEDFields as Integer, iEDField as Integer, ExString as String
   ExString = trim$(str$(ExType)) + ","
   If elem.type = MBE_Text Then
      locate_isElementAcceptable = TRUE
   End If
   If Elem.type = MBE_Text Then
      If Elem.getString (ts$, numEDFields, enterDataFlds) = MBE_Success Then
         print #1, ExString, ts$
      End If
   End If
End Function

Sub locate_elementUnacceptable
   MbeWriteStatus "Only text allowed"
End Sub

Sub locate_displayPrompts
   MbeWriteCommand "Locate Test "
   MbeWritePrompt  "Identify line"
End Sub

Sub locate_selectElementsToModify(Sourcenum as integer, ExType as integer)
   Dim elem as New MbeElement, filePos as Long, point as MbePoint, acceptPoint as MbePoint
   Dim acceptScreenPt as MbePoint, firstScreenPt as MbePoint, stat as Integer, view as Integer
   Dim acceptView as Integer, continueLocate as Integer, haveAcceptPoint as Integer, Filename as String, Filelocation as string
    continueLocate     = FALSE
   haveAcceptPoint    = FALSE
   Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
   Input #1, FileLocation
   Close #1
   Filename = "ExcavatorType_" + trim$(str$(SourceNum)) + ".txt"
   Filename = Filelocation + "\Excavate\" + Filename
   Open Filename for Append access write as #1
   Do
      Call locate_displayPrompts ()
      MbeStartLocate 0,0,0,continueLocate
      If haveAcceptPoint Or continueLocate Then
         Call MbeSendDataPoint (acceptPoint, acceptView)
         continueLocate = FALSE
         If haveAcceptPoint Then
            haveAcceptPoint = FALSE
            point = acceptPoint
            view  = acceptView
            firstScreenPt = acceptScreenPt
         End If
```

```
        If MbeState.inputType = MBE_ResetInput Then
          Exit Do
        ElseIf MbeState.inputType = MBE_DataPointInput Then
          stat = MbeState.getInputDataPoint (point, view, firstScreenPt)
          MbeSendLastInput
        End If
      End If
      If MbeState.cmdResult = MBE_AcceptQuery Then
        Do
          filePos = elem.fromLocate ()
          If locate_isElementAcceptable (elem,ExType) Then
            elem.display MBE_Hilite
            MbeWritePrompt "Accept / Reject"
            MbeGetInput MBE_DataPointInput, MBE_ResetInput
            If MbeState.inputType = MBE_DataPointInput Then
              stat = MbeState.getInputDataPoint (acceptPoint, _
                              acceptView, acceptScreenPt)
              Call locate_manipulateElement (point, view)
              continueLocate = locate_pointsCloseEnough (acceptScreenPt, _firstScreenPt)
              haveAcceptPoint = TRUE
            Else
              elem.display MBE_NormalDraw
              acceptPoint = point
              acceptView = view
              continueLocate = TRUE
            End If
            Exit Do
          Else
            MbeSendReset
            If MbeState.cmdResult = MBE_ElementNotFound Then
              Call locate_elementUnacceptable
              Exit Do
            End If
          End If
        Loop While elem.type <> MBE_Text
      ElseIf MbeState.cmdResult = MBE_ElementNotFound Then
        MbeWriteError "Element Not Found"
      Else
        print "unexpected MbeState.cmdResult of "; MbeState.cmdResult
      End If
    Loop
    Close #1
End Sub

Sub Main
    Dim saveColor as Integer, saveWeight as Integer, saveMsgs as Integer, elemSet as New MbeElementSet
    Dim message as string, SourceNum as integer, descinput as string, ExType as integer, Filename as string, FileLocation as string
    Dim exnum as integer, i as integer, Ex() as string
    Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
    Input #1, FileLocation
    Close #1
    Filename = Filelocation + "\Excavate\ExcavatorTypes.txt"
    Open Filename for Input as #1
    Line Input #1, descinput
       ExNum = val(descinput)
    Close #1
'CREATE THE ARRAY FOR EXCAVATOR TYPES
    Open Filename for Input as #1
    Redim ex(exnum)
    Line Input #1, descinput
    For i = 1 to ExNum
      Line Input #1, descinput
      Ex(i) = "Excavator Type (" + trim$(descinput) + ")"
    Next
    Close #1
    ' set MicroStation to a neutral state
    MbeSendCommand "NULL"
' get rid of any selection set
```

```
   End If
   MbeWriteStatus "Leaving locate function"
   Message = MbeInputBox ("Enter the source number: ", "1", "")
   SourceNum=Val(Message)
   ExType% = MbeSelectbox("Select Excavator Type",Ex(),"")
locate_selectElementsToModify sourcenum,ExType
' Reset settings to saved settings
   MbeSettings.color  = saveColor
   MbeSettings.weight  = saveWeight
   MbeState.messages  = saveMsgs
MbeWritePrompt
   MbeWriteCommand
End Sub
```

7) VOLUMES - MACRO TO SELECT AND EXTRACT VOLUME CALCULATIONS IN CAD AS TEXT, LEVEL 60

```
'*********************************************************************
' Basic program to display information about an element
' Copyright (1995) Bentley Systems, Inc., All rights reserved.
'  $Workfile:  elemshow.bas  $
'  $Revision:  6.6  $
'     $Date:  10 Aug 1995 15:15:58  $
' "MicroStation" is a registered trademark of Bentley Systems, Inc.
' Limited permission is hereby granted to reproduce and modify this
' copyrighted material provided that the resulting code is used only
' in conjunction with Bentley Systems products under the terms of the
' license agreement provided therein, and that this notice is retained
' in its entirety in any such reproduction or modification.
Option Base 1
Function isCell (elemType as Integer) as Integer
   If (elemType = MBE_CellLibraryHdr) Or (elemType = MBE_CellHeader) _
      Or (elemType = MBE_SharedCellDefinition) Or (elemType = MBE_SharedCell) Then
      isCell = 1
   Else
      isCell = 0
   End If
End Function
Sub ShowElemInfo (level as Integer, inElem as MbeElement)
If inElem.type = MBE_text then
   If inElem.level = 60 then
   dim range as MbeRange, origin as MbePoint, endPnt as MbePoint, cellBox() as MbePoint, points() as MbePoint
   dim cellLevels() as Integer, iPoint as Integer, iMask as Integer, indent as Integer,  ts$ as String
   dim enterDataFlds() as MbeEDField, numEDFields as Integer, iEDField as Integer
   indent = level * 2 + 10
   If inElem.getRange (range) = MBE_Success Then
      print Spc(indent);"range :";range.xLow; range.yLow; range.zLow
      print Spc(indent);"       ";range.xHigh; range.yHigh; range.zHigh
   End If
   If inElem.getOrigin (origin) = MBE_Success Then
      print Spc(indent);"Origin :";origin.x; origin.y; origin.z
   End If
   If inElem.getEndPoints (origin, endPnt) = MBE_Success Then
      print Spc(indent);"Start :";origin.x; origin.y; origin.z
      print Spc(indent);"End   :";endPnt.x; endPnt.y; endPnt.z
   End If
   If inElem.getPoints (points) = MBE_Success Then
      print Spc(indent); UBound(points) - LBound(points) + 1; " Points:"
      For iPoint = LBound(points) To UBound(points)
         print Spc(indent);"x :";points(iPoint).x;" y:";points(iPoint).y;
         print " z:";points(iPoint).z
      Next iPoint
   End If
   If inElem.type = MBE_Text Then
      If inElem.getString (ts$, numEDFields, enterDataFlds) = MBE_Success Then
         print #1, ts$; " ED Fields "; numEDFields
         If numEDFields > 0 Then
```

```
            print #1, " start ";enterDataFlds(iEDField).start;
            print #1, " length ";enterDataFlds(iEDField).length;
            print #1, " just ";enterDataFlds(iEDField).justification
          Next iEDField
        End If
      End If
    End If
    End If
  End if
  End Sub

Sub ProcessElement (nestLevel as Integer, inElem as MbeElement)
  Dim gotNext as Integer
  Do
    ' call function that shows info for element at current position
    Call ShowElemInfo (nestLevel, inELem)
    If inElem.isHeader <> 0 Then
      ' if any components in complex, process them recursively
      If inElem.nextComponent = MBE_Success Then
        Call ProcessElement (nestLevel+1, inElem)
      End If
      gotNext = inElem.nextElement
    Else
      gotNext = inElem.nextComponent
    End If
  Loop While gotNext = MBE_Success
End Sub

Sub main
  Dim element as New MbeElement, filePos as Long, Filename as string, FileLocation as string
  Open "C:\USEPWIN\Program\SimulationFile.txt" for input as #1
  Input #1, FileLocation
  Close #1
  Filename = Filelocation + "\Volumes\VolumesFromMSta.txt"
  Open Filename for Output as #1
  filePos = element.fromFile (0)
  Do While filePos >= 0
    Call ProcessElement (0, element)
  filePos = element.fromFile (filePos + element.fileSize)
  Loop
  Close #1
End Sub
```

<u>8) HAULROAD</u> - MACRO TO DEFINE THE HAUL ROADS, ROLLING RESISTANCE AND INTERSECTIONS

```
'*******************************************************************
'--------------------------------------------------------------------
' Placeroad - Based on user's points, define the road for simulation purposes
Sub WhichButton (buttonValue as long, point as Mbepoint, RollingResistance as string, IntNum as string)
  Select Case buttonValue
    Case MBE_BUTTON_OK
      'MbeMessageBox "Point was saved to file"
      X$ = format$(point.x,"###.0")
      Y$ = format$(point.y,"###.0")
      Z$ = format$(point.z,"###.000")
      Print #1, X$, Y$,Z$,trim$(RollingResistance), trim$(IntNum)
    Case MBE_BUTTON_CANCEL
      'MbeMessageBox "Cancel button was pressed, the point was not saved"
  End Select
End Sub ' WhichButton
'--------------------------------------------------------------------
' Main - entry point
Sub Main
  Dim activeColor%, activeWeight%, activeStyle%, activeStyleName$, point As MbePoint, view%, status%
  Dim errNum%, Numrecords as integer, saveMessages, descinput as string, numsource as integer
  Dim numplacement as integer, RoadSource as integer, RoadPlacement as integer, road as integer
```

```
numrecords=1
' DETERMINE THE NUMBER OF PLACEMENTS AND SOURCES FROM THE SITEDESC FILE
  Open "C:\USEPWIN\PROGRAM\SimulationFile.txt" For Input as #1
  Input #1, FileLocation
  Close #1
  FileName = FileLocation + "SiteDesc.txt"
  Open Filename for input as #1
  Line Input #1,descinput
   numsource = val(descinput)
  Line Input #1,descinput
   numplacement = val(descinput)
  Close #1
'CREATE THE ARRAY FOR PLACEMENTS
  Redim source(numsource)
  For i = 1 to numsource
    Source(i) = "Source(" + trim$(str$(i)) + ")"
  Next
'CREATE THE ARRAY FOR PLACEMENTS
  Redim placement(numplacement)
  For i = 1 to numplacement
    Placement(i) = "Placement(" + trim$(str$(i)) + ")"
  Next
'*******************************************************************
' SELECT THE ORIGIN SOURCE AND DESTINATION PLACEMENT
 DefineSource:
  roadsource% = MbeSelectbox("Select Source",Source(),"")
  If roadsource < 1 or roadsource > numsource Then
    MbeMessageBox "Please choose a defined source"
    Goto DefineSource
  End If
  roadplacement% = MbeSelectbox("Select Placement Area",placement(),"")
 DefinePlacement:
  If roadplacement < 1 or roadplacement > numplacement then
    MbeMessageBox "Please choose a defined placement"
    Goto DefinePlacement
  End If
  Message = "Describe the road using points" + Chr$(10)
  Message = Message + "from the source to the placement area"
  MbeMessageBox Message
'*******************************************************************
' OPEN THE FILE THAT DEFINES THE ROAD FROM THE APPROPRIATE SOURCE TO PLACEMENT
' FILE NAME = ROAD_"source#"_"placement#" . txt
' ie if Source=1 and Placement=2 then filename = Road_1_2.txt
 Filename$ = "\Roads\Road_" + trim$(str$(roadsource)) + "_" + trim$(str$(roadplacement)) + ".txt"
 Filename$ = Filelocation + Filename$
 Open Filename for output access write lock write as #1
'*******************************************************************
' Turn off all messages except those written by this macro
  saveMessages    = MbeState.messages
  MbeState.messages = 0
' Set up a runtime error handler
  On Error Goto handleError
' Start the default command
  MbeStartDefaultCommand
'*******************************************************************
' THIS SECTION ALLOWS THE USER TO DEFINE THE ROAD
'*******************************************************************
NumPoints = 0
  Do
'   Display messages
      MbeWriteCommand "Define Road"
      MbeWritePrompt "Enter Point"
'   Wait for a data point or a reset
      MbeGetInput MBE_DataPointInput, MBE_ResetInput
'   If a reset was entered, exit the loop
     If MbeState.inputType = MBE_ResetInput Then
         MbeWriteCommand ""
         MbeWritePrompt ""
```

```
    End IF
'   Extract the data point coordinates and view
     status = MbeState.getInputDataPoint (point, view)
     X$ = format$(point.x,"###.0")
     Y$ = format$(point.y,"###.0")
     Z$ = format$(point.z,"###.000")
     NumPoints = NumPoints + 1
     If NumPoints = 1 then RollingResistance = "0"
     If NumPoints = 1 then IntNum = "0"
     If NumPoints > 1 then
        Message = "Enter the Rolling Resistance for this section of road: "
        Message = Message + Chr$(10) + " in % "
        RollingResistance = MbeInputBox(Message,"3","")
        Message = "Enter an Intersection Number if any for the end of this section of road: "
        Message = Message + Chr$(10) + " Enter 0 for no intersection "
        IntNum = MbeInputBox(Message,"0","")
     End If
     Message = "Print this point to file?" + Chr$(10)
     Message = Message + "X= " + X + Chr$(10) + "Y= " + Y + Chr$(10) + "Z= " + Z
     WhichButton (MbeMessageBox (message, MBE_OKCancelBox)),point, RollingResistance, IntNum
   Loop
   Close #1
' Restore the message display
   MbeState.messages = saveMessages
' Start the default command--generally, Choose Element
   MbeStartDefaultCommand
   Exit Sub
handleError:
' Get the runtime error number
   errNum = Err()
' Make the macro exit with the error if it's not "Command received"
   If errNum <> MBE_ReceivedCommand Then
      Error errNum
   End If
   Exit Sub
End Sub
```

9) SOURCEROAD - MACRO TO DEFINE THE ROADS IN THE SOURCE INCLUDING ROLLING RESISTANCE

```
'**********************************************************************
' Sourcerd - Based on user's points, define the arterial road for the source
'       - Define the road from the road exiting the source into the source
'-------------------------------------------------------------
Sub WhichButton (buttonValue as long, point as Mbepoint,RollingResistance as string)
   Select Case buttonValue
      Case MBE_BUTTON_OK
         X$ = format$(point.x,"###.0")
         Y$ = format$(point.y,"###.0")
         Z$ = format$(point.z,"###.000")
         Print #1, X$, Y$,Z$,trim$(RollingResistance)
      Case MBE_BUTTON_CANCEL
   End Select
end sub ' WhichButton
'-------------------------------------------------------------
' Main - entry point
'-------------------------------------------------------------
Sub Main
   Dim saveMessages, activeColor%, activeWeight%, activeStyle%, activeStyleName$
   Dim point As MbePoint, view%, status%, errNum%, Numrecords as integer
   Dim descinput as string, descinput2 as string, numsource as integer, RoadSource as integer
   Dim RoadBench as integer, road as integer, source() as string, i as integer
   Dim benches() as string, numbenches as integer, num as single, Message as string
   Dim FileLocation as string, FileName as string, NumPoints as Integer, RollingResistance as string
  numrecords=1
' DETERMINE THE NUMBER OF SOURCES FROM THE SITEDESC FILE
   Open "C:\USEPWIN\PROGRAM\SimulationFile.txt" For Input as #1
   Input #1, FileLocation
```

```
    Open Filename for input as #1
    Line Input #1,descinput
     numsource = val(descinput)
    Close #1
'CREATE THE ARRAY FOR Sources
    Redim source(numsource)
    For i = 1 to numsource
       Source(i) = "Source(" + trim$(str$(i)) + ")"
    Next
'****************************************************************
' SELECT THE ORIGIN SOURCE
  DefineSource:
    roadsource% = MbeSelectbox("Select Source",Source(),"")
    If roadsource < 1 or roadsource > numsource Then
       MbeMessageBox "Please choose a defined source"
       Goto DefineSource
    End If
'CREATE THE ARRAY FOR Benches
    FileName$ = FileLocation + "SourceDesc"+ trim$(str$(roadsource)) + ".txt"
    Open Filename for input as #1
    Line Input #1,  descinput
    Line Input #1, descinput
    Close #1
    numbenches = val(descinput)
    Redim Benches(numbenches)
    For i = 1 to numbenches
       Benches(i) = "Bench(" + trim$(str$(i)) + ")"
    Next
    roadbench% = MbeSelectbox("Select Bench ",Benches(),"")
    Message =  "Describe the road using points" + Chr$(10)
    Message = Message + "from the source edge to the farthest extent in the source"
    MbeMessageBox Message
'****************************************************************
' OPEN THE FILE THAT DEFINES THE ROAD FROM THE APPROPRIATE SOURCE TO PLACEMENT
  Filename$ = "RoadArterialSource_" + trim$(str$(roadsource)) +"_" + trim$(str$(roadbench)) +".txt"
  Filename = filelocation + "\Roads\" + filename
  Open Filename for output access write lock write as #1
'****************************************************************
' Turn off all messages except those written by this macro
    saveMessages     = MbeState.messages
    MbeState.messages = 0
' Set up a runtime error handler
    On Error Goto handleError
' Start the default command
    MbeStartDefaultCommand
'****************************************************************
' THIS SECTION ALLOWS THE USER TO DEFINE THE ROAD
'****************************************************************
     Do
'    Display messages
        MbeWriteCommand "Define Source Arterial Road"
        MbeWritePrompt  "Enter Point"
'    Wait for a data point or a reset
        MbeGetInput MBE_DataPointInput, MBE_ResetInput
'    If a reset was entered, exit the loop
      If MbeState.inputType = MBE_ResetInput Then
            MbeWriteCommand ""
            MbeWritePrompt  ""
            Exit Do
      End IF
'    Extract the data point coordinates and view
        status = MbeState.getInputDataPoint (point, view)
        X$ = format$(point.x,"###.000")
        Y$ = format$(point.y,"###.000")
        Z$ = format$(point.z,"###.000")
        NumPoints = NumPoints + 1
        If NumPoints = 1 then RollingResistance = "0"
        If NumPoints > 1 then
```

```
      Message = Message + Chr$(10) + " in % "
      RollingResistance = MbeInputBox(Message,"3","")
    End If
    Message = "Print this point to file?" + Chr$(10)
    Message = Message + "X= " + X + Chr$(10) + "Y= " + Y + Chr$(10) + "Z= " + Z
    WhichButton (MbeMessageBox (message, MBE_OKCancelBox)),point, RollingResistance
  Loop
  Close #1
' Restore the message display
  MbeState.messages = saveMessages
' Start the default command--generally, Choose Element
  MbeStartDefaultCommand
  Exit Sub
handleError:
' Get the runtime error number
  errNum = Err()
' Make the macro exit with the error if it's not "Command received"
  If errNum <> MBE_ReceivedCommand Then
    Error errNum
  End If
  Exit Sub
End Sub
```

CASE STUDY: SIMULATION INPUT AND OUTPUT

CASE STUDY INPUT

The input to the case study, from the CAD, includes the following:

1) The road input for the case study can be found in Tables 6-3 and 6-4 in the main body of the thesis.

2) The source roads are described below in Table C-1.

Table C-1 Case Study: Source Roads

Arterial Section	Bench	X1	Y1	Z1	X2	Y2	Z2	Station	Length	Slope	Rollong Resistance
3	1	468778	6352027	291.0	468720	6351712	291.0	703.1	320.0	0.0	4
2	1	469012	6352222	291.0	468778	6352027	291.0	383.0	304.6	0.0	4
1	1	469075	6352268	298.4	469012	6352222	291.0	78.5	78.5	9.4	4
3	2	468743	6352088	283.0	468743	6351746	283.0	725.0	342.1	0.0	4
2	2	468980	6352187	283.0	468743	6352088	283.0	382.9	257.0	0.0	4
1	2	469075	6352268	298.4	468980	6352187	283.0	125.9	125.9	12.2	4
3	3	468735	6352012	275.0	468735	6351810	275.0	629.5	202.0	0.0	4
2	3	468957	6352172	275.0	468735	6352012	275.0	427.5	273.4	0.0	4
1	3	469075	6352268	298.4	468957	6352172	275.0	154.1	154.1	15.2	4

3) Table C-2 presents a sample of the extracted data describing the excavation that is input to the simulation model. The input to the simulation models below contains the data for the first 30 excavation blocks that will be excavated, as indicated by the order column. The area for each subgrid is 2500 m^2. The volumes of the soils in a block are determined from the area multiplied by the depth. The EX-3500 is used for each block, and the loading model used is the trucks and shovel loading both sides and illustrated in Figure 3-13 of the main body of the thesis.

Sequence	Bench	Order	BenchDesc	Depth	Xi	Yi	Xcoord	Ycoord	Surface	Soil
1	1	1	BENCH292	1.84	12	15	469065	6352265	1	Overburden
2	1	2	BENCH292	2.50	12	14	469065	6352215	1	Overburden
2	1	2	BENCH292	2.81	12	14	469065	6352215	2	Silt
2	1	2	BENCH292	1.78	12	14	469065	6352215	3	Clay
2	1	2	BENCH292	1.04	12	14	469065	6352215	5	Sand
3	1	3	BENCH292	2.92	11	14	469015	6352215	1	Overburden
3	1	3	BENCH292	2.65	11	14	469015	6352215	2	Silt
3	1	3	BENCH292	1.75	11	14	469015	6352215	3	Clay
3	1	3	BENCH292	1.01	11	14	469015	6352215	5	Sand
4	1	4	BENCH292	1.87	11	15	469015	6352265	1	Overburden
4	1	4	BENCH292	2.52	11	15	469015	6352265	2	Silt
4	1	4	BENCH292	1.98	11	15	469015	6352265	3	Clay
4	1	4	BENCH292	0.92	11	15	469015	6352265	5	Sand
5	1	5	BENCH292	1.84	11	16	469015	6352315	1	Overburden
5	1	5	BENCH292	0.07	11	16	469015	6352315	2	Silt
6	1	6	BENCH292	1.67	10	16	468965	6352315	1	Overburden
6	1	6	BENCH292	2.86	10	16	468965	6352315	2	Silt
6	1	6	BENCH292	2.11	10	16	468965	6352315	3	Clay
6	1	6	BENCH292	0.84	10	16	468965	6352315	5	Sand
7	1	7	BENCH292	1.59	10	17	468965	6352365	1	Overburden
7	1	7	BENCH292	0.50	10	17	468965	6352365	2	Silt
8	1	8	BENCH292	1.59	10	15	468965	6352265	1	Overburden
8	1	8	BENCH292	2.81	10	15	468965	6352265	2	Silt
8	1	8	BENCH292	1.94	10	15	468965	6352265	3	Clay
8	1	8	BENCH292	0.90	10	15	468965	6352265	5	Sand
9	1	9	BENCH292	2.45	10	14	468965	6352215	1	Overburden
9	1	9	BENCH292	2.78	10	14	468965	6352215	2	Silt
9	1	9	BENCH292	1.72	10	14	468965	6352215	3	Clay
9	1	9	BENCH292	1.00	10	14	468965	6352215	5	Sand
10	1	10	BENCH292	2.55	10	13	468965	6352165	1	Overburden
10	1	10	BENCH292	3.04	10	13	468965	6352165	2	Silt
10	1	10	BENCH292	1.50	10	13	468965	6352165	3	Clay
10	1	10	BENCH292	1.08	10	13	468965	6352165	5	Sand
11	1	11	BENCH292	2.53	11	13	469015	6352165	1	Overburden
11	1	11	BENCH292	3.19	11	13	469015	6352165	2	Silt
11	1	11	BENCH292	1.53	11	13	469015	6352165	3	Clay
11	1	11	BENCH292	1.11	11	13	469015	6352165	5	Sand
12	1	12	BENCH292	2.06	12	13	469065	6352165	1	Overburden
12	1	12	BENCH292	3.18	12	13	469065	6352165	2	Silt
12	1	12	BENCH292	1.56	12	13	469065	6352165	3	Clay
12	1	12	BENCH292	1.13	12	13	469065	6352165	5	Sand
13	1	13	BENCH292	2.23	13	14	469115	6352215	1	Overburden
13	1	13	BENCH292	0.13	13	14	469115	6352215	2	Silt
14	1	14	BENCH292	1.91	13	13	469115	6352165	1	Overburden

14	1	14	BENCH292	1.57	13	13	469115	6352165	3	Clay
15	1	15	BENCH292	1.67	13	12	469115	6352115	1	Overburden
15	1	15	BENCH292	3.51	13	12	469115	6352115	2	Silt
15	1	15	BENCH292	1.38	13	12	469115	6352115	3	Clay
15	1	15	BENCH292	0.52	13	12	469115	6352115	5	Sand
16	1	16	BENCH292	1.99	12	12	469065	6352115	1	Overburden
16	1	16	BENCH292	3.34	12	12	469065	6352115	2	Silt
16	1	16	BENCH292	1.35	12	12	469065	6352115	3	Clay
16	1	16	BENCH292	1.21	12	12	469065	6352115	5	Sand
17	1	17	BENCH292	2.07	11	12	469015	6352115	1	Overburden
17	1	17	BENCH292	3.45	11	12	469015	6352115	2	Silt
17	1	17	BENCH292	1.32	11	12	469015	6352115	3	Clay
17	1	17	BENCH292	1.18	11	12	469015	6352115	5	Sand
18	1	18	BENCH292	1.77	10	12	468965	6352115	1	Overburden
18	1	18	BENCH292	3.53	10	12	468965	6352115	2	Silt
18	1	18	BENCH292	1.29	10	12	468965	6352115	3	Clay
18	1	18	BENCH292	1.16	10	12	468965	6352115	5	Sand
19	1	19	BENCH292	2.12	9	12	468915	6352115	1	Overburden
19	1	19	BENCH292	3.41	9	12	468915	6352115	2	Silt
19	1	19	BENCH292	1.30	9	12	468915	6352115	3	Clay
19	1	19	BENCH292	0.14	9	12	468915	6352115	4	Clay
19	1	19	BENCH292	0.98	9	12	468915	6352115	5	Sand
20	1	20	BENCH292	2.48	9	13	468915	6352165	1	Overburden
20	1	20	BENCH292	3.04	9	13	468915	6352165	2	Silt
20	1	20	BENCH292	1.47	9	13	468915	6352165	3	Clay
20	1	20	BENCH292	1.06	9	13	468915	6352165	5	Sand
21	1	21	BENCH292	2.00	9	14	468915	6352215	1	Overburden
21	1	21	BENCH292	3.06	9	14	468915	6352215	2	Silt
21	1	21	BENCH292	1.70	9	14	468915	6352215	3	Clay
21	1	21	BENCH292	0.97	9	14	468915	6352215	5	Sand
22	1	22	BENCH292	1.41	9	15	468915	6352265	1	Overburden
22	1	22	BENCH292	3.09	9	15	468915	6352265	2	Silt
22	1	22	BENCH292	1.91	9	15	468915	6352265	3	Clay
22	1	22	BENCH292	0.89	9	15	468915	6352265	5	Sand
23	1	23	BENCH292	1.35	9	16	468915	6352315	1	Overburden
23	1	23	BENCH292	3.18	9	16	468915	6352315	2	Silt
23	1	23	BENCH292	2.03	9	16	468915	6352315	3	Clay
23	1	23	BENCH292	0.83	9	16	468915	6352315	5	Sand
24	1	24	BENCH292	1.24	9	17	468915	6352365	1	Overburden
24	1	24	BENCH292	3.59	9	17	468915	6352365	2	Silt
24	1	24	BENCH292	1.74	9	17	468915	6352365	3	Clay
24	1	24	BENCH292	0.68	9	17	468915	6352365	5	Sand
25	1	25	BENCH292	0.28	8	17	468865	6352365	1	Overburden
25	1	25	BENCH292	3.89	8	17	468865	6352365	2	Silt
25	1	25	BENCH292	1.66	8	17	468865	6352365	3	Clay
25	1	25	BENCH292	0.90	8	17	468865	6352365	5	Sand
26	1	26	BENCH292	0.46	8	16	468865	6352315	1	Overburden

26	1	26	BENCH292	1.95	8	16	468865	6352315	3	Clay
26	1	26	BENCH292	0.82	8	16	468865	6352315	5	Sand
27	1	27	BENCH292	1.38	8	15	468865	6352265	1	Overburden
27	1	27	BENCH292	3.37	8	15	468865	6352265	2	Silt
27	1	27	BENCH292	1.88	8	15	468865	6352265	3	Clay
27	1	27	BENCH292	0.86	8	15	468865	6352265	5	Sand
28	1	28	BENCH292	1.85	8	14	468865	6352215	1	Overburden
28	1	28	BENCH292	3.35	8	14	468865	6352215	2	Silt
28	1	28	BENCH292	1.66	8	14	468865	6352215	3	Clay
28	1	28	BENCH292	0.95	8	14	468865	6352215	5	Sand
29	1	29	BENCH292	2.00	8	13	468865	6352165	1	Overburden
29	1	29	BENCH292	3.32	8	13	468865	6352165	2	Silt
29	1	29	BENCH292	1.45	8	13	468865	6352165	3	Clay
29	1	29	BENCH292	1.03	8	13	468865	6352165	5	Sand
30	1	30	BENCH292	2.37	8	12	468865	6352115	1	Overburden
30	1	30	BENCH292	3.16	8	12	468865	6352115	2	Silt
30	1	30	BENCH292	1.46	8	12	468865	6352115	3	Clay
30	1	30	BENCH292	0.76	8	12	468865	6352115	4	Clay
30	1	30	BENCH292	0.26	8	12	468865	6352115	5	Sand

CASE STUDY OUTPUT

This Appendix presents a representative sample of the simulation output for the case study. Each line is a cycle for a truck. Each excavation block is excavated for five loads, for the purpose of this summary. By viewing the data, the variation in the ideal loading productivity is evident. The neural network determined the ideal loading productivity. The variation in activity durations, due to changing distances in the source, and the uniform distribution of randomness for representing activity durations is also evident in the data.

Table C-3 Sample Input to the Simulation Model: Case Study

HaulerNumber	TNOW (min)	Xnum	Ynum	LoadingProd (BCM/hr)	Bench	Order	RemainingVolume (BCM)	Cycle_Time (min)	Source_Cycle_Time (min)	Loading_Time (min)	Hauler_Waiting_Time (min)	Travel_Empty (min)	Travel_In_Source_Empty (min)	Travel_In_Source_Loaded (min)	Travel_Loaded (min)	Travel_In_Placement_Loaded (min)	Dump_Time (min)	Travel_In_Placement_Empty (min)
3	20.89	12	15	1357	1	1	4500	20.9	6.5	4.7	0.0	0.0	0.1	0.5	8.5	2.0	2.0	1.9
2	25.92	12	15	1357	1	1	4400	25.9	11.1	4.6	4.7	0.0	0.1	0.5	8.8	2.2	1.8	2.0
5	29.95	12	15	1357	1	1	4300	29.9	15.4	4.3	9.3	0.0	0.1	0.5	8.5	2.2	1.8	2.1
4	34.03	12	15	1357	1	1	4200	34.0	19.9	4.5	13.6	0.0	0.1	0.5	8.2	2.0	2.1	1.9
6	38.74	12	15	1357	1	1	4100	38.7	24.5	4.6	18.1	0.0	0.1	0.5	8.2	2.0	2.0	2.1
1	44.09	12	14	1661	1	2	20225	44.1	30.0	3.7	24.2	0.0	0.1	0.8	8.6	1.8	1.8	1.8
3	48.13	12	14	1661	1	2	20125	27.2	6.3	3.8	0.5	6.5	0.1	0.7	8.7	2.0	1.8	2.0
2	52.50	12	14	1661	1	2	20025	26.6	5.4	3.4	0.0	6.4	0.2	0.7	8.6	2.1	2.0	2.0
5	56.21	12	14	1661	1	2	19925	26.3	5.3	3.3	0.0	6.7	0.1	0.7	8.3	1.8	2.2	2.0
4	60.10	12	14	1661	1	2	19825	26.1	5.5	3.4	0.0	6.4	0.1	0.7	8.3	1.9	1.9	2.1
6	66.35	11	14	1664	1	3	20725	27.6	6.6	3.9	0.2	6.4	0.1	1.1	8.4	2.1	2.1	2.0
1	71.53	11	14	1664	1	3	20625	27.4	6.5	3.8	0.0	6.5	0.5	1.1	8.4	2.0	1.8	2.2
3	75.59	11	14	1664	1	3	20525	27.5	6.4	3.5	0.0	6.6	0.5	1.2	8.4	1.9	2.1	2.1
2	79.96	11	14	1664	1	3	20425	27.5	6.4	3.5	0.0	6.4	0.5	1.2	8.4	2.1	2.2	1.9
5	84.38	11	14	1664	1	3	20325	28.2	6.3	3.5	0.0	7.3	0.4	1.1	8.6	2.1	2.0	1.9
4	88.73	11	15	1594	1	4	18125	28.6	7.9	3.6	2.0	6.4	0.4	0.7	8.5	1.9	1.8	2.0
6	93.93	11	15	1594	1	4	18025	27.6	6.1	4.0	0.0	6.8	0.1	0.7	8.5	2.2	2.0	2.1
1	98.09	11	15	1594	1	4	17925	26.6	5.7	3.7	0.0	6.5	0.1	0.7	8.4	2.1	1.8	2.0
3	101.60	11	15	1594	1	4	17825	26.0	5.5	3.4	0.0	6.5	0.1	0.8	8.4	1.8	1.9	1.9
2	106.76	11	15	1594	1	4	17725	26.8	6.1	4.1	0.0	6.5	0.1	0.7	8.2	2.0	2.0	2.0
5	112.74	11	16	1329	1	5	4675	28.4	7.8	4.2	1.4	6.3	0.1	0.8	8.3	1.8	2.0	2.1
4	118.02	11	16	1329	1	5	4575	29.3	8.1	4.8	1.0	6.5	0.2	0.9	8.3	2.0	2.2	2.1
6	122.11	11	16	1329	1	5	4475	28.2	7.0	4.4	0.3	6.8	0.2	0.9	8.4	2.0	1.8	2.1
1	126.92	11	16	1329	1	5	4375	28.8	7.9	4.7	0.8	6.6	0.2	1.0	8.5	1.9	1.8	2.1
3	131.93	11	16	1329	1	5	4275	30.3	8.6	4.5	1.8	6.8	0.2	0.9	8.9	2.2	1.8	2.0
2	137.77	10	16	1605	1	6	18600	31.0	9.3	4.1	2.7	6.7	0.2	1.1	8.7	2.0	2.2	2.1
5	140.97	10	16	1605	1	6	18500	28.2	7.5	3.8	1.2	6.3	0.2	1.1	8.5	1.9	1.9	2.1
4	145.47	10	16	1605	1	6	18400	27.5	6.2	3.7	0.0	6.4	0.2	1.1	9.0	2.0	1.8	2.0
6	149.34	10	16	1605	1	6	18300	27.2	6.4	3.9	0.0	6.6	0.2	1.1	8.4	1.9	2.1	1.9
1	154.04	10	16	1605	1	6	18200	27.1	6.1	3.5	0.0	6.4	0.2	1.1	8.5	2.2	2.1	1.9
3	159.89	10	17	1349	1	7	5125	28.0	7.4	4.4	0.0	6.4	0.2	1.5	8.3	1.9	1.9	2.1
2	165.39	10	17	1349	1	7	5025	27.6	7.1	4.3	0.0	6.4	0.3	1.4	8.2	2.0	1.8	2.1
5	170.78	10	17	1349	1	7	4925	29.8	8.0	4.3	0.9	6.6	0.3	1.3	8.9	2.1	2.1	2.2
4	174.07	10	17	1349	1	7	4825	28.6	8.1	4.3	0.9	6.4	0.3	1.4	8.2	1.9	2.1	2.0
6	179.06	10	17	1349	1	7	4725	29.7	8.7	4.7	1.1	6.6	0.3	1.3	8.5	2.2	1.8	2.0
1	185.36	10	15	1591	1	8	18000	31.3	9.4	3.6	2.8	6.5	0.2	1.6	9.2	1.9	2.2	2.0
3	188.42	10	15	1591	1	8	17900	28.5	6.7	3.6	0.0	7.3	0.5	1.4	8.6	2.0	2.1	1.8
2	193.20	10	15	1591	1	8	17800	27.8	6.7	3.6	0.0	6.4	0.5	1.4	8.5	2.0	2.0	2.2
5	199.32	10	15	1591	1	8	17700	28.5	7.2	4.1	0.0	6.6	0.5	1.4	8.7	2.1	2.0	1.9

6	208.73	10	14	1629	1	9	19775	27.7	7.8	3.7	1.1	6.4	0.5	1.1	8.0	1.8	2.0	1.8
1	213.07	10	14	1629	1	9	19675	27.7	6.4	3.5	0.0	6.4	0.4	1.3	9.1	1.9	1.9	2.1
3	215.69	10	14	1629	1	9	19575	27.3	6.9	3.4	0.5	6.3	0.5	1.4	8.1	2.0	2.0	1.9
2	221.51	10	14	1629	1	9	19475	28.3	6.9	3.9	0.0	6.6	0.5	1.4	8.4	2.1	2.1	2.1
5	228.04	10	14	1629	1	9	19375	28.7	6.9	3.9	0.0	6.3	0.5	1.3	9.0	2.2	2.1	2.2
4	232.42	10	13	1639	1	10	20325	29.9	8.5	3.6	1.9	6.7	0.5	1.3	8.9	1.9	2.1	1.9
6	235.74	10	13	1639	1	10	20225	29.0	8.0	4.0	1.2	6.8	0.4	1.1	8.5	2.1	1.8	1.8
1	240.24	10	13	1639	1	10	20125	27.2	6.4	3.6	0.0	6.5	0.5	1.1	8.4	1.9	1.8	2.1
3	244.38	10	13	1639	1	10	20025	28.7	7.5	3.9	0.9	6.7	0.5	1.1	8.3	2.0	2.1	2.1
2	248.54	10	13	1639	1	10	19925	27.0	6.2	3.5	0.0	6.7	0.4	1.1	8.3	1.9	2.1	1.8
5	256.33	11	13	1647	1	11	20800	28.3	6.6	3.6	0.0	6.7	0.5	1.3	8.9	1.8	2.0	2.2
4	260.50	11	13	1647	1	11	20700	28.1	6.7	3.6	0.0	6.8	0.5	1.4	8.8	2.0	1.9	1.9
6	263.20	11	13	1647	1	11	20600	27.5	7.1	3.4	0.7	6.4	0.5	1.3	8.1	2.0	2.1	1.8
1	268.70	11	13	1647	1	11	20500	28.5	7.0	4.0	0.0	6.6	0.6	1.3	8.6	2.1	2.0	2.2
3	271.89	11	13	1647	1	11	20400	27.5	6.9	3.6	0.2	6.2	0.5	1.3	8.2	1.9	2.2	2.1
2	276.67	12	13	1658	1	12	19725	28.1	7.1	3.5	0.8	6.6	0.5	1.0	8.3	1.9	2.1	2.2
5	283.95	12	13	1658	1	12	19625	27.6	6.3	4.0	0.0	6.6	0.2	1.0	8.5	2.1	2.0	2.1
4	287.37	12	13	1658	1	12	19525	26.9	5.7	3.4	0.0	6.6	0.2	0.9	8.5	2.1	2.1	1.9
6	290.27	12	13	1658	1	12	19425	27.1	6.5	3.4	0.8	6.5	0.2	0.9	8.2	1.8	2.0	1.9
1	295.65	12	13	1658	1	12	19325	27.0	5.9	3.5	0.0	6.6	0.2	1.0	8.6	1.9	1.9	2.1
3	304.19	13	14	1350	1	13	5800	32.3	11.7	4.1	1.9	6.4	0.2	4.2	8.4	1.8	1.9	2.1
2	309.25	13	14	1350	1	13	5700	32.6	11.6	4.6	0.1	6.4	1.4	4.3	8.7	1.9	2.1	2.0
5	316.17	13	14	1350	1	13	5600	32.2	11.2	4.9	0.0	6.5	1.3	3.9	8.4	2.0	2.1	2.0
4	320.87	13	14	1350	1	13	5500	33.5	11.9	4.1	1.6	6.4	1.3	3.8	8.9	2.1	2.2	2.1
6	325.60	13	14	1350	1	13	5400	35.3	14.6	4.8	2.9	6.3	1.2	4.5	8.3	2.0	2.1	2.0
1	326.98	13	13	1646	1	14	16425	31.3	10.3	3.4	3.2	6.9	1.4	1.1	8.5	1.9	1.8	2.0
3	331.86	13	13	1646	1	14	16325	27.7	6.0	3.4	0.0	6.6	0.2	1.2	8.9	2.2	2.2	1.8
2	337.01	13	13	1646	1	14	16225	27.8	6.6	3.9	0.0	6.6	0.2	1.2	8.9	1.9	1.8	1.9
5	343.16	13	13	1646	1	14	16125	27.0	6.1	3.5	0.0	6.4	0.2	1.2	8.2	2.2	2.0	2.0
4	348.46	13	13	1646	1	14	16025	27.6	6.5	3.9	0.0	6.6	0.2	1.1	8.4	1.9	2.2	2.0
6	354.42	13	12	1640	1	15	17600	28.8	7.4	3.7	0.8	6.5	0.2	1.5	8.6	2.1	2.0	2.2
1	357.42	13	12	1640	1	15	17500	30.4	8.9	3.8	2.3	7.3	0.3	1.3	8.4	1.9	2.2	1.8
3	361.61	13	12	1640	1	15	17400	29.8	9.1	3.9	2.3	6.3	0.3	1.5	8.4	1.9	2.0	2.1
2	365.65	13	12	1640	1	15	17300	28.6	7.1	3.5	0.7	6.7	0.3	1.4	8.8	2.0	1.9	2.2
5	371.49	13	12	1640	1	15	17200	28.3	6.7	3.9	0.0	7.0	0.3	1.3	8.5	2.1	2.0	2.0
4	377.21	12	12	1626	1	16	19625	28.8	7.2	3.7	0.4	6.7	0.3	1.7	8.8	2.0	2.0	2.0
6	383.20	12	12	1626	1	16	19525	28.8	7.3	3.6	0.0	6.7	0.6	1.9	8.7	2.1	2.2	1.9
1	386.25	12	12	1626	1	16	19425	28.8	8.4	3.8	1.0	6.3	0.5	2.0	8.1	1.9	2.1	2.0
3	390.53	12	12	1626	1	16	19325	28.9	7.1	3.6	0.0	7.3	0.6	1.7	8.2	2.1	2.1	2.1
2	394.38	12	12	1626	1	16	19225	28.7	7.7	3.7	0.4	6.5	0.6	1.7	8.8	1.9	2.0	1.9
5	398.99	11	12	1632	1	17	19950	27.5	6.7	3.4	0.0	6.5	0.6	1.5	8.5	2.0	1.8	2.1
4	405.06	11	12	1632	1	17	19850	27.8	7.0	3.4	0.0	6.5	0.6	1.7	8.6	1.8	1.9	2.1
6	411.02	11	12	1632	1	17	19750	27.8	6.9	3.6	0.0	6.6	0.5	1.5	8.3	2.1	1.8	2.0
1	415.42	11	12	1632	1	17	19650	29.2	7.8	3.4	1.0	6.2	0.6	1.7	9.1	2.0	2.0	2.1
3	419.05	11	12	1632	1	17	19550	28.5	7.3	4.0	0.0	6.6	0.6	1.5	9.0	1.8	2.0	1.8
2	424.28	10	12	1619	1	18	19275	29.9	8.6	4.0	1.4	6.8	0.6	1.4	8.5	2.0	1.8	2.2
5	427.97	10	12	1619	1	18	19175	29.0	7.7	3.5	1.1	6.5	0.5	1.4	8.6	2.0	2.1	1.9
4	433.93	10	12	1619	1	18	19075	28.9	7.0	3.8	0.0	7.1	0.5	1.4	8.7	2.1	1.9	2.1
6	439.43	10	12	1619	1	18	18975	28.4	6.8	3.7	0.0	6.3	0.5	1.4	9.1	1.8	2.2	2.2
1	442.82	10	12	1619	1	18	18875	27.4	6.9	3.8	0.0	6.3	0.5	1.4	8.2	2.0	2.0	1.9
3	448.08	9	12	1658	1	19	19775	29.0	8.0	3.6	1.6	6.5	0.5	1.2	8.6	2.1	2.0	1.9
2	451.88	9	12	1658	1	19	19675	27.6	6.9	3.9	0.1	6.3	0.5	1.2	8.5	1.8	2.0	2.1
5	455.34	9	12	1658	1	19	19575	27.4	6.3	3.4	0.0	6.6	0.5	1.2	8.5	2.0	2.1	1.9

6	467.92	9	12	1658	1	19	19875	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	473.06	9	13	1660	1	20	20025	30.2	8.8	3.8	2.0	6.6	0.4	1.3	8.6	2.1	2.1	2.1
3	476.16	9	13	1660	1	20	19925	28.1	6.8	3.5	0.6	6.6	0.5	1.1	8.9	1.8	1.9	2.0
2	479.43	9	13	1660	1	20	19825	27.5	6.8	3.5	0.4	6.4	0.5	1.2	8.2	1.9	2.0	2.2
5	483.09	9	13	1660	1	20	19725	27.7	7.0	4.0	0.2	6.6	0.5	1.2	8.1	2.1	1.9	2.0
4	487.84	9	13	1660	1	20	19625	27.1	6.5	3.6	0.0	6.5	0.4	1.2	8.1	1.9	2.1	2.0
6	496.02	9	14	1654	1	21	19225	28.1	6.8	3.7	0.0	6.9	0.5	1.4	8.5	2.0	2.1	1.8
1	501.13	9	14	1654	1	21	19125	28.1	6.9	3.8	0.0	6.8	0.5	1.3	8.4	2.1	2.1	1.8
3	504.95	9	14	1654	1	21	19025	28.8	7.6	3.6	0.8	6.6	0.6	1.4	8.5	1.9	2.2	2.0
2	508.26	9	14	1654	1	21	18925	28.8	8.2	3.7	1.5	6.4	0.5	1.4	8.5	1.9	1.8	2.0
5	512.40	9	14	1654	1	21	18825	29.3	8.1	3.7	1.3	6.6	0.5	1.4	8.4	2.1	2.1	2.0
4	518.39	9	15	1594	1	22	18150	30.6	9.1	3.9	1.7	6.6	0.6	1.8	8.6	2.2	1.9	2.1
6	524.76	9	15	1594	1	22	18050	28.7	7.6	4.0	0.0	6.3	0.6	1.8	9.0	1.9	1.8	2.1
1	529.79	9	15	1594	1	22	17950	28.7	7.4	4.1	0.0	6.4	0.5	1.5	8.4	2.2	2.1	2.2
3	533.05	9	15	1594	1	22	17850	28.1	7.2	3.9	0.0	6.8	0.6	1.6	8.3	1.9	1.9	2.0
2	538.09	9	15	1594	1	22	17750	29.8	8.3	4.1	0.6	6.8	0.6	1.8	8.6	2.1	1.9	2.2
5	543.60	9	16	1647	1	23	18375	31.2	10.0	3.8	2.4	6.4	0.6	2.0	8.4	2.2	2.0	2.2
4	547.33	9	16	1647	1	23	18275	28.9	7.7	3.7	0.1	6.4	0.6	1.9	8.6	2.2	2.0	2.1
6	553.30	9	16	1647	1	23	18175	28.5	7.7	3.9	0.0	6.2	0.6	1.9	8.7	1.9	2.2	1.9
1	558.09	9	16	1647	1	23	18075	28.3	7.2	3.4	0.0	6.6	0.6	2.0	8.6	2.1	1.9	1.9
3	562.82	9	16	1647	1	23	17975	29.8	7.9	4.0	0.0	7.1	0.7	2.1	8.8	2.0	2.0	2.0
2	567.65	9	17	1644	1	24	18025	29.6	8.4	3.7	1.1	6.4	0.6	1.7	8.6	1.9	2.1	2.2
5	571.75	9	17	1644	1	24	17925	28.1	7.0	3.9	0.0	6.7	0.3	1.5	8.4	2.1	1.8	2.1
4	575.71	9	17	1644	1	24	17825	28.4	7.1	3.8	0.1	6.9	0.3	1.7	8.6	2.1	1.9	1.8
6	580.73	9	17	1644	1	24	17725	27.4	6.9	3.8	0.0	6.5	0.3	1.6	8.2	2.0	1.9	1.9
1	585.37	9	17	1644	1	24	17625	27.3	6.7	3.6	0.0	6.2	0.3	1.6	8.4	2.2	2.0	1.8
3	591.50	8	17	1629	1	25	16725	28.7	7.8	3.5	0.2	6.4	0.3	2.6	8.7	2.0	2.0	1.8
2	596.59	8	17	1629	1	25	16625	28.9	7.8	3.6	0.0	6.6	0.7	2.3	8.6	2.0	1.9	2.1
5	600.23	8	17	1629	1	25	16525	28.5	7.8	3.6	0.0	6.5	0.8	2.2	8.4	1.8	2.0	2.0
4	604.59	8	17	1629	1	25	16425	28.9	7.9	3.6	0.0	6.6	0.8	2.3	8.2	2.2	2.0	2.1
6	609.66	8	17	1629	1	25	16325	28.9	8.2	4.0	0.0	6.2	0.8	2.2	8.5	1.8	2.1	2.1
1	614.37	8	16	1629	1	26	16725	29.0	7.9	3.5	0.6	6.5	0.7	1.9	8.6	2.1	1.9	2.1
3	620.43	8	16	1629	1	26	16625	28.9	7.5	3.5	0.0	6.3	0.7	2.2	8.8	2.1	2.1	2.1
2	625.29	8	16	1629	1	26	16525	28.7	7.7	4.0	0.0	6.7	0.6	1.9	8.2	1.9	2.1	2.2
5	628.83	8	16	1629	1	26	16425	28.6	8.1	3.4	0.6	6.4	0.6	2.3	8.3	1.9	1.8	1.9
4	633.51	8	16	1629	1	26	16325	28.9	7.3	3.4	0.0	6.8	0.7	2.0	8.9	2.1	1.9	1.9
6	638.14	8	15	1650	1	27	18625	28.5	7.6	3.8	0.2	6.4	0.6	1.8	8.7	1.9	1.9	1.9
1	642.30	8	15	1650	1	27	18525	27.9	6.9	3.4	0.0	6.5	0.6	1.7	8.6	2.0	1.9	2.2
3	649.23	8	15	1650	1	27	18425	28.8	7.3	3.6	0.0	6.5	0.6	1.8	8.7	2.0	2.1	2.1
2	653.92	8	15	1650	1	27	18325	28.6	7.4	3.7	0.0	6.6	0.6	1.9	9.0	1.8	2.0	1.8
5	657.31	8	15	1650	1	27	18225	28.5	7.8	3.9	0.2	6.5	0.6	1.9	8.3	2.0	1.9	1.9
4	662.98	8	14	1622	1	28	19425	29.5	7.9	3.7	0.7	6.7	0.6	1.7	8.6	2.0	2.0	2.2
6	666.76	8	14	1622	1	28	19325	28.6	6.9	3.5	0.0	6.6	0.6	1.7	8.7	2.2	2.2	2.0
1	671.17	8	14	1622	1	28	19225	28.9	7.2	3.8	0.0	6.8	0.6	1.6	8.4	2.1	2.2	2.2
3	677.50	8	14	1622	1	28	19125	28.3	7.5	4.0	0.0	6.4	0.5	1.7	8.6	2.1	1.9	1.8
2	682.03	8	14	1622	1	28	19025	28.1	7.3	3.8	0.0	6.5	0.6	1.7	8.4	1.8	2.0	2.1
5	686.80	8	13	1656	1	29	19400	29.5	8.6	3.8	1.6	6.7	0.6	1.4	8.3	2.1	1.9	1.8
4	691.07	8	13	1656	1	29	19300	28.1	6.9	3.7	0.1	6.5	0.5	1.4	8.3	2.2	2.0	2.2
6	694.92	8	13	1656	1	29	19200	28.2	7.0	3.8	0.1	6.5	0.5	1.3	8.6	2.0	2.0	2.0
1	699.14	8	13	1656	1	29	19100	28.0	7.0	3.6	0.0	6.5	0.6	1.5	8.5	2.0	2.1	2.0
3	705.52	8	13	1656	1	29	19000	28.0	6.6	3.6	0.0	6.5	0.5	1.3	9.0	2.0	2.0	1.9
2	710.38	8	12	1659	1	30	19925	28.3	7.1	3.8	0.4	6.6	0.5	1.1	8.6	1.8	2.2	2.0
5	713.94	8	12	1659	1	30	19825	27.1	6.2	3.5	0.0	6.5	0.4	1.1	8.5	1.9	1.9	2.1

APPENDIX D

INTELLIGENT DATA MANAGER

The intelligent data manager consists of three components. The first is the interface to start the simulation. The second is to evaluate data extracted from the CAD files, perform calculations and store the data in a database. The third function is to control the direction of the simulation by linking the simulation to the database. The third function includes simulation output. The second and third functions use Visual Basic and a database, and are not contained in this Appendix.

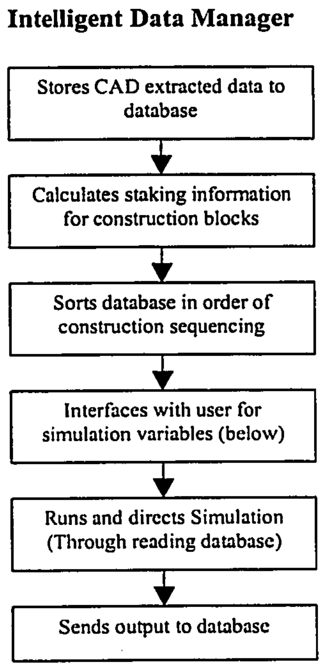


Figure D-1 Intelligent Data Manager

Simulation Interface

The simulation interface begins by displaying to the user the first interface screen. From the main form, the user moves to forms for entering data.

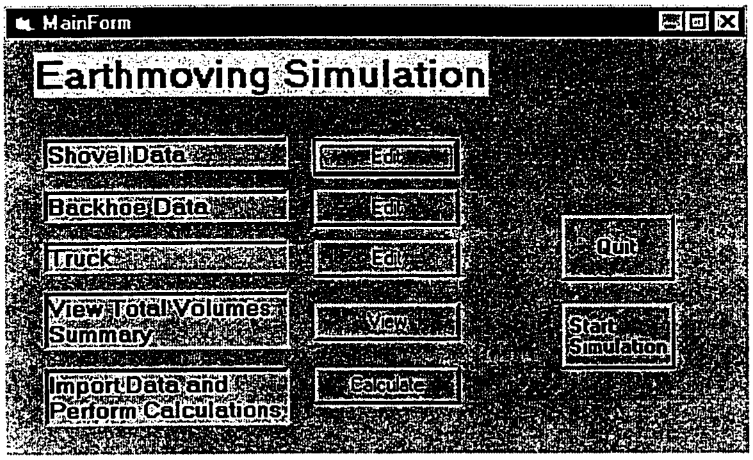


Figure D-2 Interface: Main Form

The shovel data form shown in Figure D-2 illustrates the form used to enter specific durations for the truck activities at the shovel loading position. The simulation models are:

Simulation Type 1: Shovel and trucks, loading both sides

Simulation Type 2: Shovel and trucks, loading one side

Simulation Type 3: Shovel and trucks, drive-by loading

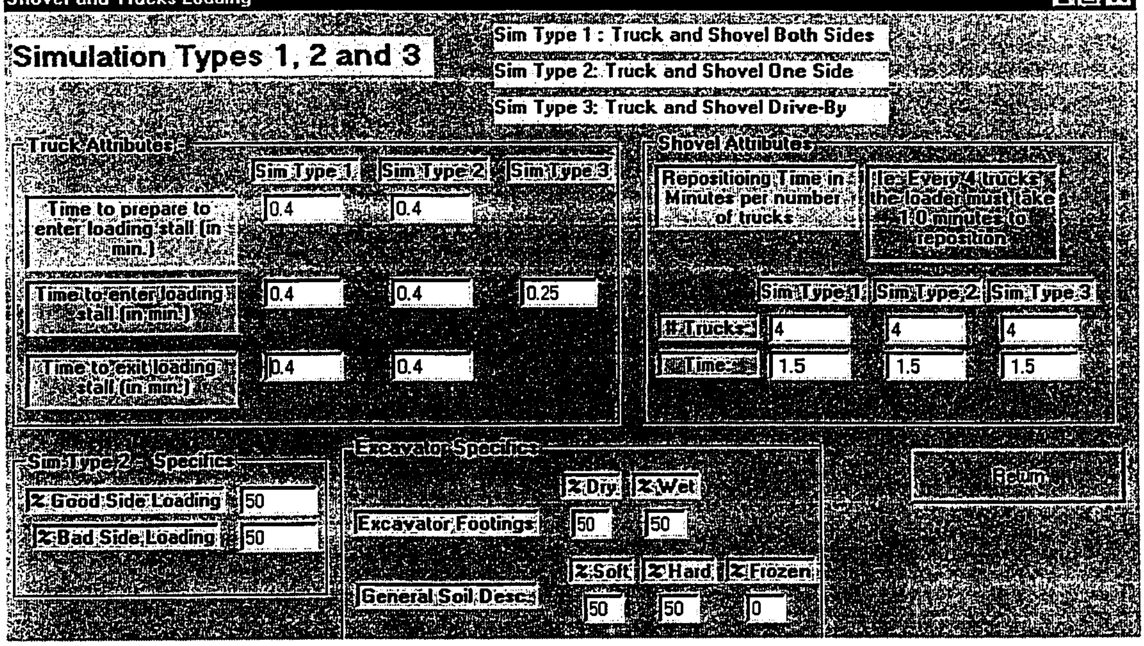


Figure D-2 Interface: Shovels and Trucks Form

The backhoe and trucks form is used to define the truck activities at the loading operations for the backhoe and trucks. The different simulation models are:

Simulation Type 4: Backhoe and trucks, top-loading

Simulation Type 5: Backhoe and trucks, drive-by loading

Simulation Type 6: Backhoe and trucks, level loading

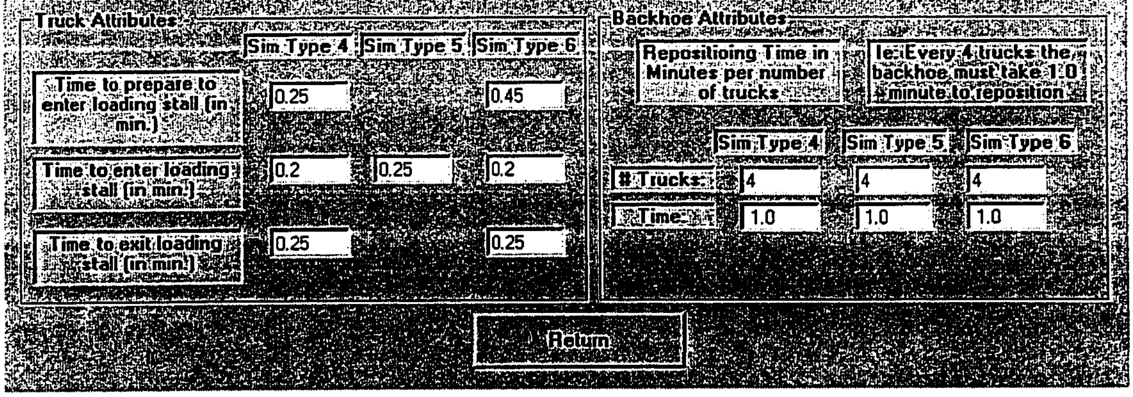


Figure D-4 Interface: Backhoe and Trucks Form

The trucks form is used to define the number of trucks in the system and the trucks capacity, as described in Figure D-4.

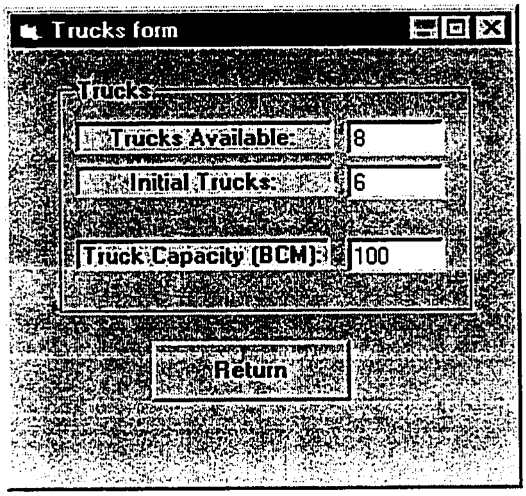
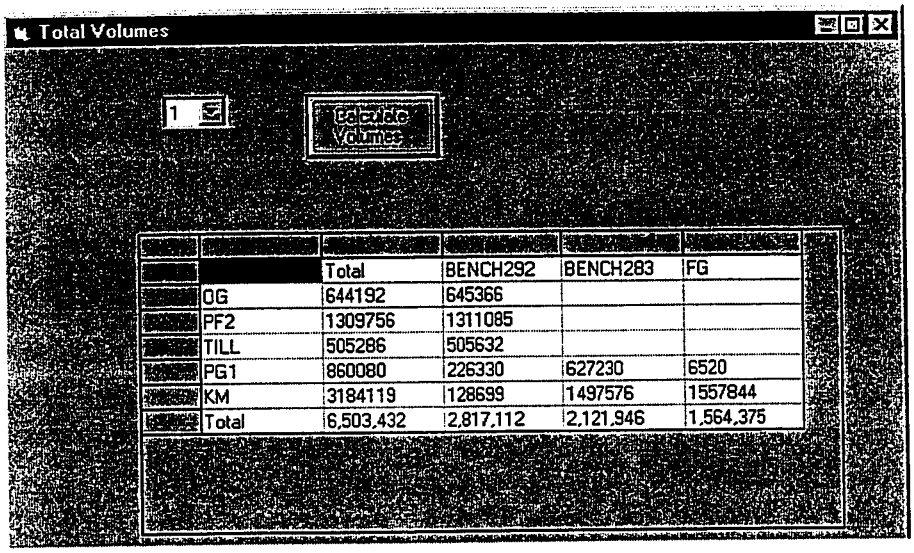


Figure D-4 Interface: Trucks Form

excavation, as illustrated in Figure D-5.



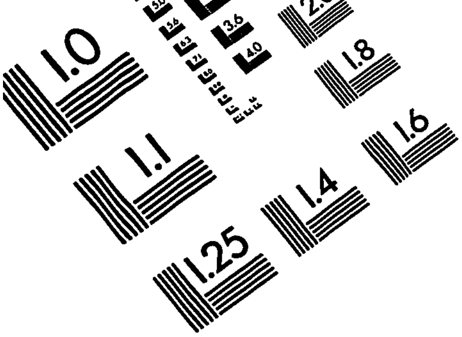
Total Volumes

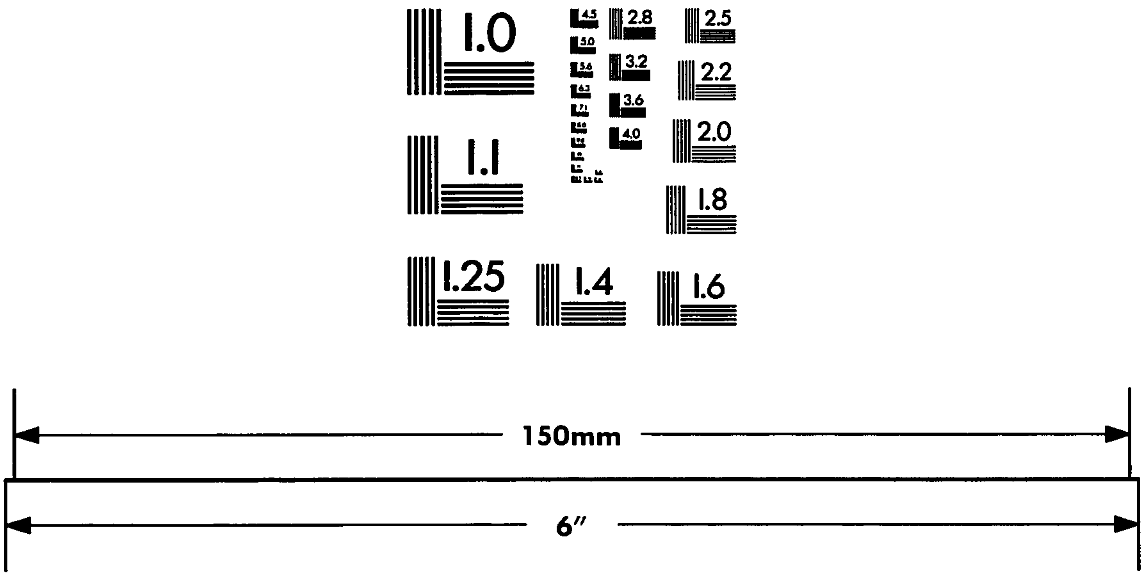
1

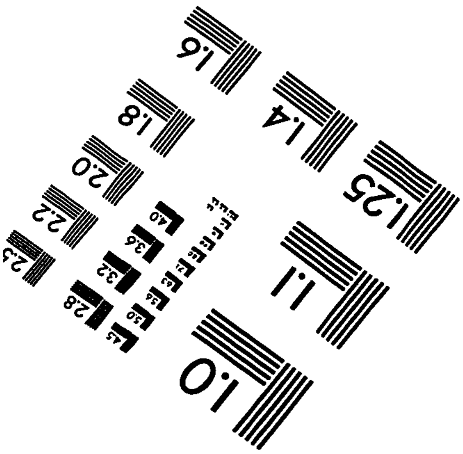
Calculate Volumes

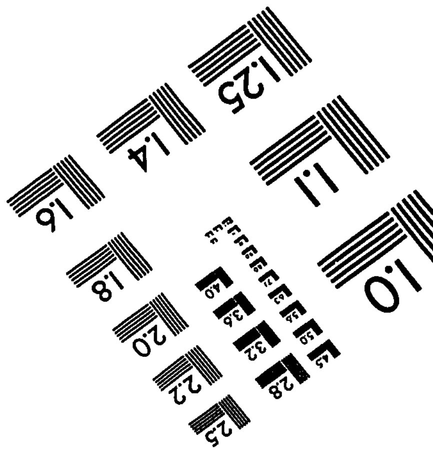
	Total	BENCH292	BENCH283	FG
OG	644192	645366		
PF2	1309756	1311085		
TILL	505286	505632		
PG1	860080	226330	627230	6520
KM	3184119	128699	1497576	1557844
Total	6,503,432	2,817,112	2,121,946	1,564,375

Figure D-5 Interface: Volumes Calculations Form









APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989