Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

INTEGRATING NEURAL NETWORKS WITH SPECIAL PURPOSE SIMULATION

Dany Hajjar
Simaan AbouRizk
Kevin Mather

Department of Civil Engineering
220 Civil Electrical Building
University of Alberta
Edmonton CANADA T6G 2G7

ABSTRACT

Traditional methods of dealing with variability in simulation input data are mainly stochastic. This is most often the best method to use if the factors affecting the variation or the nature of the relationships between the factors and the outputs can not be easily identified.

Artificial Neural Networks have the ability to learn complex relationships between inputs and outputs. Their use can greatly enhance simulation models and allow for more accurate representations of real life scenarios.

This paper proposes a generic approach for integrating external processes such as neural networks with simulation models. The object oriented method is used to 'expose' the properties of the simulation models to external processes, and allow for users to define relationships at run-time.

This approach was tested by integrating a neural network model for predicting the productivity of an excavator with an earth-moving simulation process. This proved to be of extreme benefit because the defined neural network parameters depend on certain factors which varied during the simulation.

1 INTRODUCTION

Input modeling for simulation studies plays a significant role in the effectiveness of model accuracy. The input modeling process begins with the collection of sample data. This data is then analyzed to determine the best input model. If the data range is relatively small, a deterministic approach is used; the average value is used as as direct input to the simulation. If the range is large, statistical analysis packages such as ExpertFit (Averill 1998) can be used to obtain a stochastic distribution that best models the data. When the model uses stochastic distributions, the simulation experiment should be repeated a number of times to obtain an accurate statistical distribution for the results.

Stochastic modeling is often the only option when the factors affecting the variation or the relationship between the factors and the output cannot be identified. However, ignoring the cause of this variation can make the results of the simulation misleading, especially if this variation is due to a parameter that dynamically changes within the simulation model.

If the factors affecting the variation can be determined, a different approach based on Artificial Neural Networks (ANN) can be utilized. ANN's are effective tools capable of learning complex relationships by examining a known set of data. They are extremely effective in learning, even in the presence of small errors in the sample data.

One method of integrating neural networks with simulation models is to simply use a separate program such as NeuralWorks (NeuralWare 1998) to develop a neural network, provide it with the desired input values, obtain the outputs, and use outputs as parameters to the simulation program. However, this procedure cannot be used if the input values of the neural network change during the simulation.

A generic approach for integrating simulation models and external systems such as neural networks is required. This approach should be able to encapsulate these external processes and provide standard access methods for exchanging information with the simulation models.

Such an approach was designed and implemented to integrate an earth moving simulation program called AP2Earth with a neural network model for predicting excavator productivity. A more detailed overview of neural networks and some background on AP2Earth is provided first.

2 BACKGROUND

2.1 Neural Network Modeling

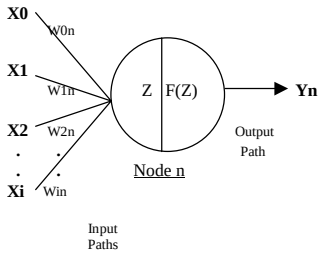

Figure 1: Example Neural Network Node

Neural networks are a form of artificial intelligence which are 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 (Figure 1). 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 2.

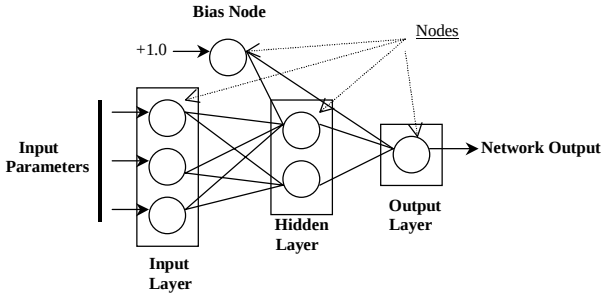

Figure 2: Example Neural Network Structure

2.2 Special Purpose Simulation for Construction Earth Moving Operations

Earth moving is a specialized construction field where large quantities of earth are moved from one location (generally referred to as the "cut"), to an another location (the "fill"). Examples of such projects include overburden removal for mining operations and construction of earth filled dams. Earth moving projects consist of many interacting processes, including preparation, excavation, hauling, dumping and spreading. Preparation is used if the earth is not suitable for immediate loading and requires ripping. Excavation is the process of transporting earth from the prepared earth pile into incoming trucks. This is done using loaders, shovels or backhoes. Hauling involves trucks traveling through roads with varying slopes and ground conditions as well as traffic intersections in order to transport earth and return to the excavation site. Dumping is the transfer of earth from the trucks into a spreading pile. This pile is spread by a number of dozers as part of the spreading process.

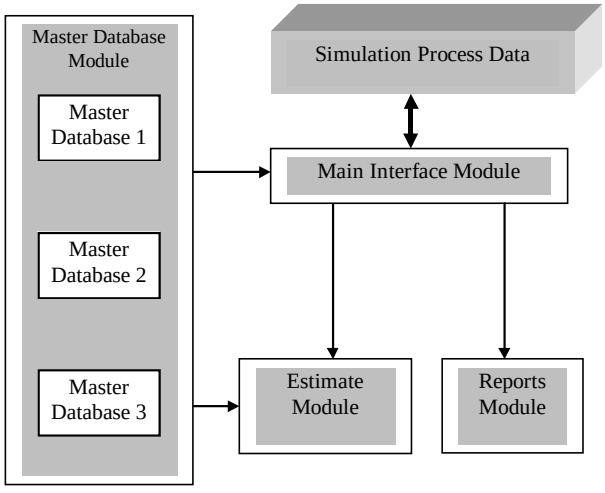

Figure 3: AP2-Earth Modules

AP2-Earth is a Special Purpose Simulation (SPS) system for the analysis of earth moving operations. It was developed with the objective of providing earth moving contractors with a system that can automate the traditionally time consuming process of manually preparing earth moving estimates for large and complex projects. Interested readers are referred to Hajjar and AbouRizk (1998) for more detailed discussion on the internal design and development aspects of AP2-Earth.

To ensure the tool's acceptance, it was built with the capability to integrate with other modules including equipment inventory databases and estimating programs. These supporting modules were also developed and provided with the main simulation module. A graphical depiction of the overall AP2-Earth modules and how they relate is illustrated in Figure 3.

The master database module consists of standard equipment and labour specifications that are available for use by the contractor. The database stores various equipment properties, some of which can be used directly in the simulation environment or at a later stage for other purposes. The main interface module is where simulation processes modules are built and manipulated. Equipment data can be imported from the master database modules. Two other modules are used to examine the simulation results. The first is the reports module, which provides various types of information ranging from high level

production reports to detailed equipment performance and utilization reports. The other is the estimate module, which allows users to develop estimates and link certain inputs directly to the simulation results.

3 INTEGRATION FRAMEWORK

Integrating simulation with other systems involves a careful analysis of the nature of the information that needs to be exchanged. Ideally, the integration mechanism should be generic enough to be used to link simulation processes to neural network models as well as many other elements such as standard equipment databases and CAD models. The authors' objective behind this research is to prototype integration strategies on AP2-Earth which can eventually be generalized to other simulation systems. A graphical illustration of the long term objective is presented in Figure 4.

The main challenges in achieving this goal lie in the means of implementing the arrows that represent the information flow between the systems. One way to implement this is to develop specialized filters that transform data between each set of processes. However, this method requires a new "filter" module to be developed for every new process or simulation package. The alternative is to develop a general standard for information exchange among such systems. This way, new processes which are developed confirming to this standard will achieve integration easily.

The authors propose an approach that utilizes the object oriented method to provide both a common interfacing mechanism for all the processes and the ability to define general linking structures between these processes. For the purposes of this paper, a process is defined as a specialized component that encapsulates a certain concept. This includes simulation processes such as excavation, neural network models, CAD models, stochastic distributions, and others. The distinctive element of a process is that it accepts a certain set of inputs, performs a certain transformation and returns the output. The inputs and outputs of a given process will be referred to as a property of that process. Properties can hold different types of data and can represent parameters or results of the internal analysis of the process.

The first step is the development of a method to represent processes generically. This means that the main properties of a given process, including its input and output properties, must be 'exposed' to the outside world through a standard interface. The second step was the development of a general purpose 'link' concept used to define the interfaces or means of transferring information between the processes.

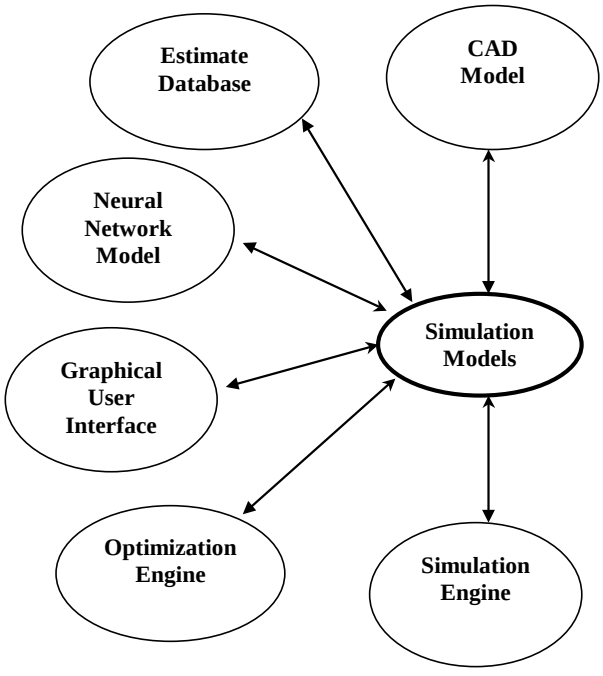

Figure 4: Integrating Simulation with Supporting systems

Note that the internal structure of each process is not restricted in any manner. The internal representation is still manipulated using specialized editors. For example, the special purpose simulation tool is still used to edit the actual simulation processes.

In the context described, the overall environment could be represented with a set of processes and links. An illustration of a general environment is shown in Figure 5. Note that the internal links between the simulation processes themselves are not shown because they require more complicated types of links, which were discussed in the Ap2Earth paper (Hajjar and AbouRizk 1998). Examples of global level properties for a simulation model include the current simulation time, number of free resources, and average process production for all processes.

Information in the form of simple data values flow in one direction along each link. The approach followed does not limit the number of links that are defined for each exposed property. Instead, link priorities are defined to determine which link should be used in case of multiple links.

One interesting side effect of this representation is the ability to represent the graphical dialog boxes as processes. Dialog boxes are used to manipulate the process properties directly by users. Links between processes and dialog

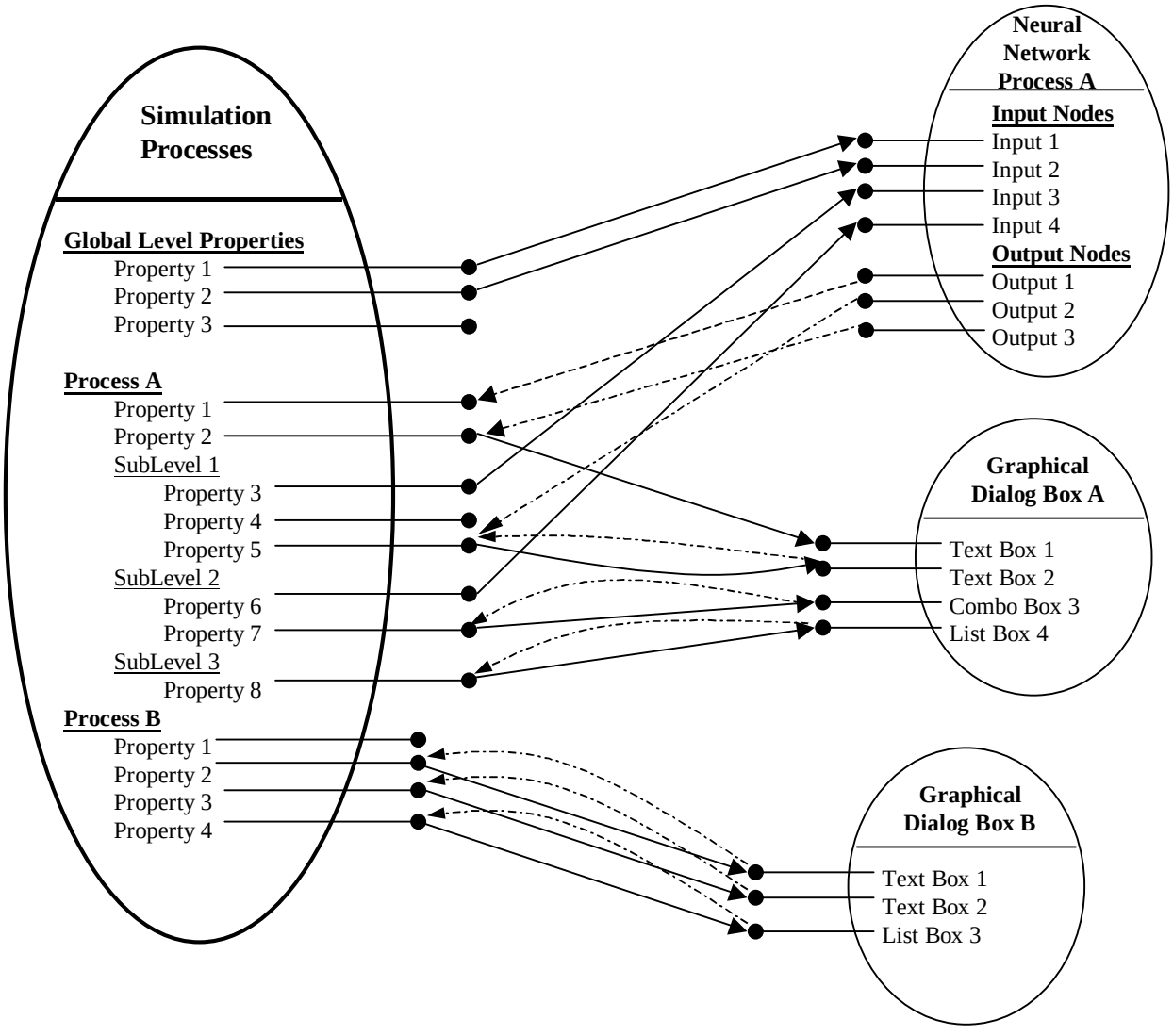

Figure 5 - Sample Integration Scenario between Simulation Processes , Graphical Dialog Boxes and Neural Networks

boxes are assigned top priority. This allows the simulation environment to act as an independent system without the need to interface with any other system. Users have the option of utilizing the default dialog box links to set the process properties manually or build new links to relate certain properties to other processes.

The implementation details will not be discussed in any great details in this paper. The main tools utilized include Microsoft Visual C++, which is an object oriented development environment. Run-time interfacing between processes is done using the OLE (Object Linking and Embedding) technology. OLE makes it possible for code components to communicate at run-time rather than hard-coding the structure at design-time. Each process (except the simulation processes themselves) was written as an OLE component.

3.1 Designing Processes For Integration

A standard method of exposing process attributes was developed. Each process contains a set of properties that are used as part of the process's internal analysis, either as inputs or outputs. These properties are defined generically in a structure that is available for other systems to examine. A general representation of this structure is shown in Table 1.

The interface type column indicates what type of access that any linked process can posses. Certain properties required by the underlying transformation code (simulation model in case of a simulation process) are set as 'Read/Write'. Other properties, which are generated by the process and cannot be modified externally, are marked as 'Read Only'. This includes such properties as output node values for a neural network or the current simulation time for a given simulation process.

Table 1 Sample Exposed Attribute Structure

Property Name	**Data Type**	**Default Value**	**Interface Type**	**Parent Node**
Property 1	Integer	10	Read Only	Node 1
Property 2	Char	"KM"	Read/Write	Node 2
…				

To facilitate navigation across the properties of a given process, it was decide to allow for the representation of properties as part of a hierarchy. This allows users to use 'drill-down' techniques to find the desired properties, instead of having to search a single list. This is accomplished using the 'Parent Node' column which is a reference to another structure called the 'Node Hierarchy Representation' (a sample is shown in Table 2). Properties can be assigned as children of a node and nodes can also be assigned as children of other nodes. This information is used at run-time to present a hierarchial navigation view in the form of a tree.

Table 2. Representation of Node Hierarchy

Node Name	**Parent Node**
Node 1	NULL
Node 2	Node 1
Node 3	Node 1
Node 4	Node 3
Node 5	Node 4

3.2 The Integration Mechanism

Once standard structures for exposing the process properties are defined, general linking structures can be defined to allow for the transfer of data from a source property in one process to a destination property in another. In object oriented terminology, a class can now be defined to hold references to (1) the source property, (2) the destination property, and (3) the link priority. The link priority is used to resolve multiple links that are connecting to a single destination property.

Algorithms were developed to allow simulation processes to retrieve the values of exposed properties. These were implemented as member functions of the developed classes. The algorithm for obtaining the actual value of a given output property in any external process is summarized as follows :

CExposedProperty:GetValue() :

1. Find all input type properties of the external process (e.g.: neural network model input nodes).
2. For each property found:
2.1 Find all links that reference the property as destination properties
2.2 Sort the found links by their assigned priority (highest priority first)
2.3 For each link:
2.3.1 Set the value of the destination property based on the value of the source property. If the source property along the current link is not defined go to the next link
2.4 If the value is still not set (all links resulted in undefined values), then set the property value to its default value.
2.5 Once all the input type properties have been set, initiate the processes transformation function (e.g.: The recall function of the neural network model)
2.6 The transformation function sets the value of the output property directly.

The implemented algorithm includes considerations for cashing the values of the output properties so that the analysis is only performed when values of the input properties change.

4 EXCAVATION CASE STUDY

The presented approach for integrity was not implemented at the full scope described. Instead, the authors have used it on a small scale to generalize AP2Earth's representation of the excavation simulation process and allow it to integrate with a neural network model to improve prediction of the loading duration.

4.1 Neural Network Process Development

A neural network was designed, developed and trained for the prediction of the excavation productivity, which determines the loading duration.

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.

4.1.1 Problem Definition

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

4.1.2 Data Collection

To test the applicability of 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 next step was to decide what data will be collected for the neural networks. An informal survey of project managers, superintendent, and foreman was conducted to determine the factors that affect 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 include the type of soils excavated, general site conditions, loading equipment utilization, and the relative positioning of the excavator and trucks.

Upon completion of the data collection, the inputs and outputs for the proposed network were identified. The construction site personnel identified four categories of factors that affect the ideal production: (1) Soil classification, (2) General soil description, (3) Excavator footing conditions, and (4) Excavator and truck relative positioning.

4.1.3 Network Development

The identified model input factors are as follows:

Soil Classification: Five generic types of soil were defined for the data collection: Overburden, silt, sand, clay, and rock.

General soil conditions: The general soil conditions are either soft, hard or frozen. 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.

Excavator footing conditions: The excavator footing conditions are either wet or dry. Wet footing conditions require more work when the excavator is repositioning and also more help from the dozer in cleaning up the areas before repositioning. Productivity of the excavator is theoretically higher for dry footing conditions.

Excavator and truck relative positioning: The excavator truck relative positioning includes four factors. 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 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.

The developed neural network model contains 13 input nodes, with one hidden layer containing 4 intermediate nodes, and one output node, the predicted productivity. The input nodes were determined from the collected data. Other factors such as whether it was a day or night shift and the identify of the foreman and 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.

4.1.4 Network Training

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, twenty percent of the data points were randomly chosen and used for network testing, while the remaining eighty percent of the points were used for data training.

The learning occurs by adjusting the weighted connections between the neurons of each slab so that the overall error between the predicted and actual output is minimized. 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.

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.

4.1.5 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 not yet encountered. The data points reserved for testing are used in the network to test acceptability of the recall network.

The average error between the productivity predicted by the network and the actual productivity is 5.6%. The maximum absolute error is 43%. This indicates that the proposed neural network is successful at predicting productivity.

4.2 Excavation Simulation Process Development

The excavation simulation process was re-designed to allow it to integrate with the neural network model. The exposed properties included the required inputs to the simulation model. It is important to mention that the soil type, which is an important input parameter of neural network model, varies during the simulation. This variation is defined by dividing the excavation site into grids and assigning a soil type to each grid. Table 3 displays a list of the exposed excavation process properties and their purposes.

4.3 Integration of the Developed Processes

The appropriate links were defined to integrate the neural network model properties with the exposed properties of the simulation process. Links with the standard excavation dialog box were assigned the top priority, followed by the links to the neural network model. This allowed the simulation to proceed without the need for a neural network model. However, in case a neural network model is used, the graphical dialog box components for the linked properties (text boxes, list boxes, etc) were left undefined so that the secondary links could be utilized by the GetValue() algorithm. A representation of the development is shown in Figure 6.

5 CONCLUSIONS

An approach has been presented to allow for the integration of simulation systems and general neural network models. The approach was designed to allow for future integration with other types of systems as well. The links between these systems is defined by users at run-time.

The benefits of such an approach include the ability to create external supporting systems that can exchange information with simulation models. Integration of a neural network model with the excavation simulation process allowed for more accurate representation of the real life operation as complex relationships between the site conditions and the excavator productivity were accurately modeled.

Table 3. Exposed Properties of the Excavation Simulation Process

Property Name	**Interface Type**	**Descripton**
Excavator Type	Read/Write	EX-1800, EX-3500, etc.
Soil type	Read/Write	Sand, Clay, Silt, etc.
Generic Soil Description	Read/Write	Soft, hard, etc.
Excavator Footing Description	Read/Write	Wet, dry
Loading Position	Read/Write	% Good Side, %Bad Side, …
Excavator Productivity	Read/Write	Cubic meters per minute
Face Height	Read/Write	Meters
Swing Angle	Read/Write	Degrees
Total earth to excavate	Read/Write	Cubic meters
Total earth excavated	Read Only	Cum. amt. of excavated earth loaded so far. this value can be used by future neural networks that take into account the equipment utilization.

ACKNOWLEDGMENTS

This project was funded by the Natural Science and Engineering Research Council of Canada under grant number IRCPJ 195558-96.

REFERENCES

Hajjar, D. and AbouRizk, S.M. (1998). "Design and Implementation of a Simulation Framework for Earth Moving Operations", Submitted to the Journal of Construction Engineering and Management, under review.

NeuralWare (1998). Neural Works Professional. Detailed information available on **http:\\www.neuralware.com**

Averill (1998). ExpertFit Distribution Fitting Software. Detailed information on **http:\\www.averill-law.com**

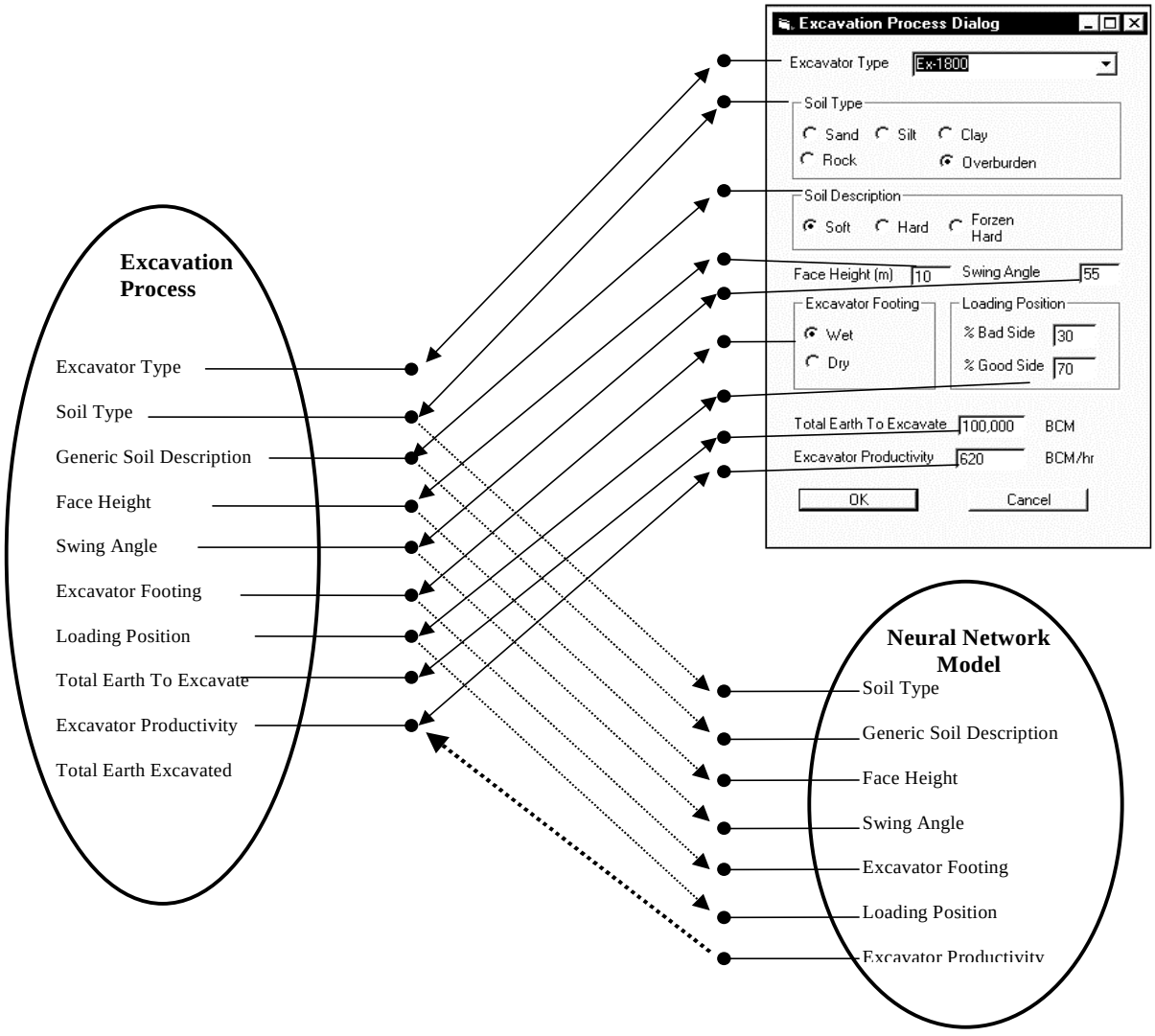

Figure 6 - Integration Excavation Simulation Process, Excavation Dialog Box and Developed Neural Networks

AUTHOR BIOGRAPHIES

DANY HAJJAR is a Ph.D. candidate in Civil Engineering at the University of Alberta. He received his Bachelor of Science with specialization in Computing Science from the University of Alberta in 1995. His research focuses on applying information modeling and object oriented methodologies to the management and control of construction projects.

SIMAAN ABOURIZK is a Professor in the Department of Civil Engineering at the University of Alberta. He received his BSCE and MSCE in Civil Engineering from Georgia Institute of Technology in 1984 and 1985, respectively; and his Ph.D. degree from Purdue University in 1990. His research interests focus on the application of computer methods and simulation techniques to the management of construction projects.

KEVIN MATHER is a project coordinator with North American Construction Group in Fort McMurry, Alberta. He graduated with a Masters of Science in Construction Engineering and Management from the University of Alberta in May of 1998. His thesis was on the topic of integration of site specific information for roadworks and strip mining into computer simulation. He received a Bachelor of Science in Civil Engineering through the Cooperative Program from the University of Alberta in 1996.