Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

BUILDING A SPECIAL PURPOSES SIMULATION TOOL FOR EARTH MOVING OPERATIONS

Dany Hajjar
Simaan AbouRizk

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

ABSTRACT

This paper discusses the process of building a special purpose simulation environment. By making the environment specific for a given industry many advantages are gained including wider acceptance and use in practical settings. Special purpose simulation tools help bring simulation to the desks of construction engineers who have little or no experience with simulation theory. By isolating the user from the low level constructs and presenting a modeling interface that more closely represents the actual system, the overall environment becomes more effective. Also, through specialization, the simulation reports can be created in the standard format that engineers in the intended domain are accustomed to. Various reports can be made to include varying levels of information summarized in a manner that can be used directly for estimating and bidding purposes. This paper shares some of the design and implementation issues used when building special purpose simulation tools.

1 INTRODUCTION

A simulation tool can be conveniently described as composed of three components: a pre-processor, a simulation algorithm (we will refer to that as "simulation engine") and a post-processor as depicted in Figure 1. The pre-processor allows for the modeling of a scenario using user level constructs involving some form of a user interface. The other function of the pre-processor is to convert the model into a format that can be used by the simulation engine. The simulation engine then performs the required analysis by performing the required number of runs, generating raw results and statistics to output files. The post-processor's main function is to read the raw information and generate the required statistics which are grouped and displayed in organized user level reports.

Traditional construction simulation tools strive to be general in order to be flexible and comprehensive.

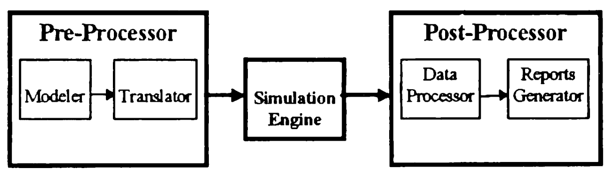

Figure 1: Simulation Components

This allows for the modeling and analysis of a wide range of application areas. While these tools are very powerful, they require the user to enter and manipulate information in a counter-intuitive manner. Construction engineers who have little experience with simulation theory must use abstract simulation level elements during the modeling phase. The correspondence between real system and the computer model is rarely similar. Also, the simulation results from these tools usually require some added level of analysis to get the necessary information. This translates into lost time and productivity. Special purpose tools remedy these deficiencies by presenting a highly visual preprocessor with a graphical user interface (GUI). This interface is tailored to the specific needs of the application domain and the end users.

Researchers in construction have used object oriented programming (OOP) to enhance the process interaction modeling construct while preserving the graphical modeling strategy of process interaction systems (e.g. COOPS (Liu and Ioannou, 1992); and CIPROS (Odeh et. al. 1992)). Although one can argue that the level of effort associated with model development greatly depends on the modeler's background (i.e. whether more effort is associated with object oriented simulation (OOS) versus traditional graphical process interaction modeling) advantages of OOS are accentuated when masses of information must be handled (e.g. modeling an entire construction project or a complex operation). The one-to-one correspondence between the model and the physical entity available through OOS may be appealing and prove very competitive as long as the modeled process is within a reasonable level of detail. Modeling more complex projects may require significant model abstraction, however, which may necessitate the use of some form of process interaction modeling elements.

The advantages of OOS include 1) reduced development time due to re-usability of model components as outlined by Odeh et.al, (1992), and 2) modularity and hierarchical structuring which would facilitate focus on high level models while maintaining the accurate representation at the process level.

Ziegler (1987) described the use of OOP as basis for modularity in simulation applications. Luna (1991, and 1992) presented concepts pertaining to hierarchical modeling for simulation. Model definition is achieved through a successive combination of two types of building blocks, the atomic model component and the coupled model component. The hierarchical structuring of the model is achieved through successive combinations of atomic and coupled models. The essence of the modeling strategy is based on input/output links between the various component. Luna uses the Smalltalk language (Goldberg and Robson 1983) to implemented the derived concepts.

2 DESIGN METHODOLOGIES

Building a special purpose simulation tool involves the design and implementation of the components described in Section 1.

The works of Ziegler and Luna formed the basis of the work described in this paper. The main focus of the described developments were on the modeling side since simulation algorithms are abundant in the literature.

In general, information about a given system is represented through processes and their interactions within the overall system. In this approach, a project is broken down into independent processes each with a distinct function. These processes interact together though shared and well defined interaction points. By modularizing these processes and defining how they can interact, a library of processes can be built, manipulated and extended to suit the needs of construction engineers. Once linked to iconic representations we achieve our objective of a flexible specialty tool for a given industry

3 BUIDLING PROCESS

This section describes the various design stages of a special purpose simulation tool.

3.1 Preliminary Conceptual Design

The design process begins with identifying the general processes that make up the entire project. The initial problem statement, the description of the high level processes and their relationships define the of the simulation tool. The next step is to list the various active and passive elements of each process. Active elements are those that cause a transition in the state of the system; they request and free up resources and drive the simulation experiment. Passive elements, on the other hand, do not actively participate in the experiment.

The sequence of events for active elements can be defined through high level flow diagrams. The flow diagrams will form the initial conceptual simulation model. they determine the required events, properties and methods for each process.

Any particulars, assumptions, and constraints are then stated as part of each process. Constraints determine how and when the simulation events will take place, for example. The assumptions and constrains will further limit the scope of the application domain.

3.1.1 Interaction Points

After the definition of each process, the nature of the interaction and relationships between the processes are conceptualized. The final simulation model will consists of a set of processes interacting in some manner. There are generally two types of interactions: production based and entity based.

Production based interactions are those that involve a product quantity being manipulated. Some processes will increase the amount of this product while others will decrease it. The dependency and, therefore, the interaction comes from the fact that the active entities within each process will wait if their actions could result in a production level that is below the minimum level or above maximum level. If the two processes are not "production compatible" (i.e. they use different measures or require different forms of production), a transfer function is defined. For example, when one process interacting with another while one is producing a product with a cubic measure while the other producing a product with a surface measure require the application of a cubic to square conversion function.

In entity based interactions, active elements pass from one process to another. Again, conversion function on the attributes of the entities might need to be specified in order to make the two processes "entity compatible".

The conceptual design stage is completed when all other elements falling outside of all the individual processes are defined. These include any global resources that are shared between processes or external interactions that can affect more than one process at a time (examples are weather processes and breakdown processes).

3.2 Simulation Level Design

At this stage, the models needed to drive the simulation of each process are formally described using entity flow diagrams. These diagrams will serve as the basis for implementing the models using discrete event simulation libraries. They are drawn from the simple flow diagram defined for each active element in the previous steps. These diagrams must be specific enough to be easily translated.

3.3 Data Structure Design

Data structure design involves the definition of the data objects or classes that will represent the elements and processes of the tool. A preliminary class hierarchy is first defined and the functionality of each one class is described. These structures will represent:

- Project level information such as global resource and container structures.
- Process level information.
- Linking information between processes.
- Active and passive element representation

This structure is then revised to implement specialized behavior requirements. For simulation support, basic extended functionality includes integrity checking, simulation initialization, processing and clean-up. Graphical user interaction behavior is then added to support the creation of a GUI.

3.4 Preprocessor Design

The graphical interface simplifies the task of constructing and running a simulation model. However, users are still working with low level simulation constructs such as resources, interaction points, and processes. As stated earlier, the objective of a special purpose simulation tools is to allow end users such as project managers or even on site engineers to use the tools effectively. This means that there is a need to add another layer of abstraction that brings the modeling environment to an intuitive and direct level. A preprocessor must be designed to take advantage of the assumptions and fundamentals of the model and reduce the required amount of input data.

3.4 Post Processor Design

The post processor objective is to analyze and present the raw simulation results in a familiar and immediately useful format. This is best done with in simple database environment where data can be quickly analyzed using queries. The final results can then be formatted and displayed in customizable database reports.

4 EXAMPLE APPLICATION: EARTH MOVING SIMULATION

Simulation studies are well suited for the analysis of earth moving for many reasons including repetition of given operations, dynamics of resource interactions, external factors that need to be included in the analysis such as weather and the general randomness associated with such systems. The ultimate objective is to build a highly visual and flexible modeling environment where a wide range of scenarios can be analyzed with a great deal of ease. The purpose of this example is not to detail the lowest level of programming or design specifications; rather it will focus on design methodologies discussed earlier and how it is applied.

4.1 Preliminary Conceptual Design

Earth moving operations involve a number of hauling trucks loading earth material from a preparation area and transporting it to a dump location. Other supporting activities that also take place are earth preparation at the source and spreading at the placement location. The entire operation can be represented as a number of interacting processes: Preparation, loading, hauling, dumping and spreading.

4.1.1 Preparation

Preparation is the process of breaking up and collecting soil so that it can be easily loaded into trucks. It involves a number of dozers cycling between the collection area and the loading pile. With each cycle, the amount of prepared earth in the loading pile increases. Preparation is not necessary if the earth is adequate for immediate loading .

4.1.2 Loading

During the loading process, several trucks wait in line to be loaded with earth from the loading pile. Several loaders could be present at one loading location. Loading durations are usually obtained through on site studies or from equipment handbooks. Waiting will result when loaders' production is less than that of the trucks or when loading production is greater than the preparation production.

4.1.3 Hauling

The hauling process involves a number of trucks hauling earth from the preparation area or "source" to the destination or "placement". Roads are normally modeled as road segments with properties such as grade, length and maximum travel speed. The haul route could contain various intersections and junctions where trucks must stop and give way to other traffic which might have higher priority. The trucks follow a static path from the source to the placement known as the "travel loaded" path and return on the "travel empty" path. The source and placement in both paths are the same therefore trucks always follow a closed loop. Different trucks could follow different paths since earth moving operations could involve several sources and placements as shown Figure 2.

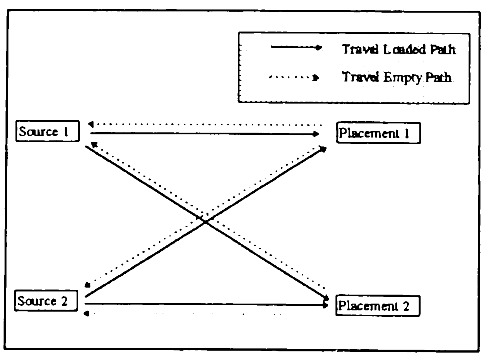

Figure 2: Maximal Path Arrangement Between Two Sources and Two Placements

The travel times of trucks can be obtained from empirical travel time curves. These curves give an approximate travel duration given the length and total effective grade of the road segment for both loaded and empty trucks. Total effective grade is composed of grade resistance and rolling resistance. Grade resistance is a measure of the force that must be overcome to move trucks over uphill slopes. Rolling resistance is a measure of the force that must be overcome to roll or pull a wheel off the ground.

Within earth moving projects, intersections serve to regulate the traffic between arriving entities. Trucks must stop at intersections and check before proceeding. Intersection are not simple resources since the maximum number of trucks or other types of traffic allowed at any one time is variable and dependent on the "crossing path" of entities. As illustrated in Figure 3, Case 1 presents no conflict between the crossing behavior of incoming traffic. The intersection serves as a relay station and no waiting is involved. On the other hand, In Case 2 traffic could queue at A or B since the desired paths cross.

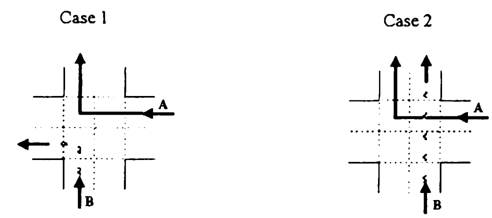

Figure 3: Two Crossing Scenarios

4.1.4 Dumping

Dumping is a simple process whereby trucks wait for a position at the dump location, then dump their payload into the spreading pile.

4.1.5 Spreading

The spreading process is the reverse of the preparation process. One or more dozers spread the hauled earth from the dump pile across a given area.

4.1.6 Interaction Points

The processes described so far are well contained and independent. To construct a useful scenario, they must be linked to form a representative model of the whole project. Production based interactions are present between preparation and loading as well as dumping and spreading. Entity based interaction points are present between loading and hauling, and between hauling and dumping. The entities that are being transferred between the two processes represent trucks.

4.1.6.1 Production Based Interactions

The preparation and dumping processes interact through the loading pile. On one side, dozers in the preparation area continually increase the level in the pile. While, on the other side, trucks are being loaded using that same pile. Queuing is possible only from the loading side as trucks will wait if not enough earth is present. A "swell" ratio is a conversion factor used in order to make the preparation and the preparation processes production compatible. This is because during preparation, quantities are tracked in *bank* cubic meters; while in loading, they are tacked as a process to a another. The choice of destinations is specified by the source process when a request for a transfer is made.

4.1.7 Sample Scenario

An example project is shown in Figure 4. It consists of several possible preparation processes, loading, hauling, dumping, and spreading processes.

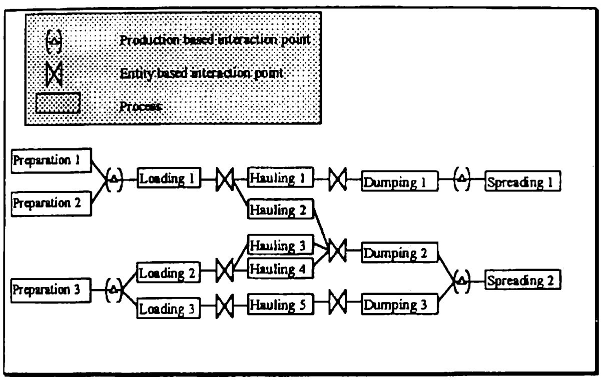

Figure 4: Example Earth Moving Project

4.2 Simulation level design

The event flow diagrams of each process can now be designed. These diagrams define the event sequence of each active element and can be directly translated into discrete-event simulation models. The symbols used are explained in Table 1. The following subsections present sample diagrams for the loading and hauling processes.

4.2.1 Loading

The loading simulation module is illustrated in Figure 5. Duration A and B represent loader holdup time and loading time consecutively. Interaction point 1 is the loading pile can be manipulated by other loading or preparation processes (see Figure 4). Interaction points 2 and 3 represent entity transfer points from and to the hauling process.

Table 1: Model Design Symbols

Symbol	Description
Event Name	Represents a simulation event or transition in simulation.
Cond A → Dur A	Represents an activity with a defined duration and a crietieria in the case of conditional branching
Resource	Allocated or releases a certain resource.
I	Accumulate a production level for any desired purpose. Used to track the cumulative production level.
+ Int. 1	Represents a production interaction points.
→ Int. 1	Represents an entity interaction point. The arrow indicates whether entities are arriving to the process or exiting from it.
* →	Indicates the starting event of a process.

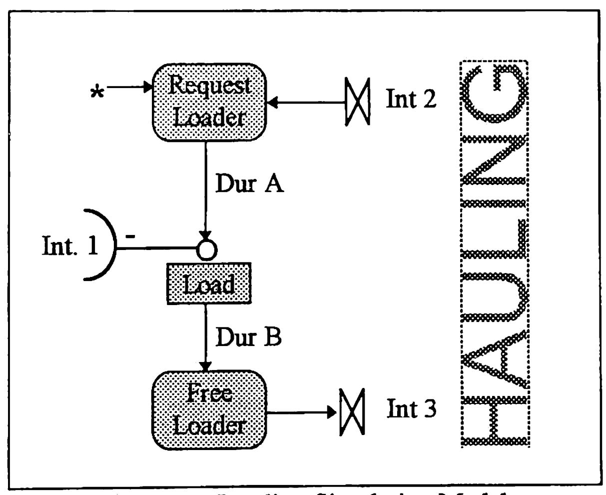

Figure 5: Loading Simulation Model

4.2.2 Hauling

The hauling simulation module is illustrated in Figure 8. Duration descriptions are shown in Table 2, Conditions in Table 3 and interaction points in Table 4.

Table 2: Durations For Hauling Model

Duration	Description
A	Segment Travel Time
B	Intersection Wait Time
C	Crossing time
D	Junction Wait Time

Table 3: Conditional Branching For Hauling

Cond.	Description
1	Travel Empty path complete
2	Next object in path is a junction
3	Next object in path is an intersection
4	Travel Loaded path complete
5	Actual hauled amount < Desired hauling amount
6	Actual hauled amount ≥ Desired hauling amount

Table 4: Interaction Information for Hauling Model

Int.	Interaction Point	Interaction Command
1	Truck departure to loading area	transfer(destination loading process)
2	Truck arrival from loading area	
3	Truck departure to dumping area	transfer(destination dumping process)
4	Truck arrival from dumping area	

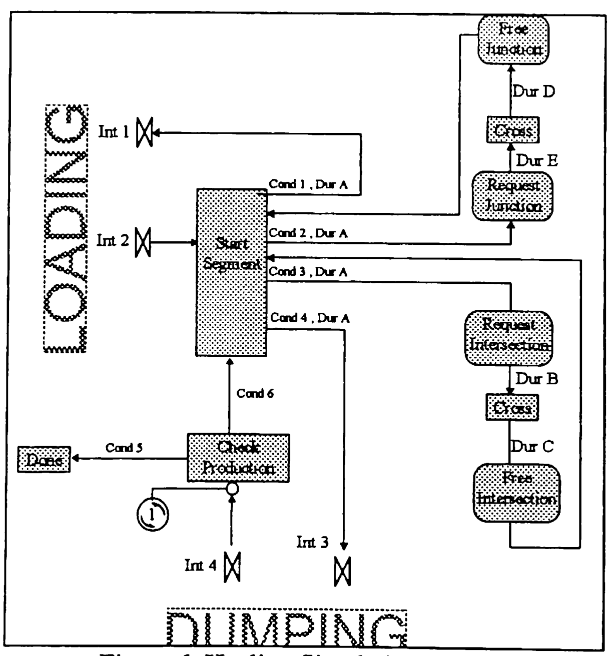

Figure 6: Hauling Simulation Model

4.2.2.1 Intersection Modeling

As explained before, intersection modeling is not a simple task. The traffic behavior at an intersections depends on several factors including priority, "crossing path" and crossing duration of incoming traffic. One way of modeling this behavior is to use a state grid. The grid determines whether a certain traffic block is currently busy or free. Traffic entities arriving at an intersection cross only when all the blocks constituting the crossing path are free; otherwise, they will queue up. When an entity crosses the intersection and releases the blocks, waiting entities are allowed to proceed based on priority. If priorities are equal, they are served on first come first serve basis. The result is a sort of four way intersection.

4.3 Data Structure Design

The class hierarchy is presented in Figure 9. It is developed through continuous refinement until all the required behaviors and objects have been represented. Following is a brief overview of some of the base classes:

4.3.1 CGraphicalBase

Implements basic graphical state and position representation and manipulation functionality.

4.3.2 CInteractivePositioning

"click and drag" interactive positioning functionality is supported through this class Classes which can take advantage of this behavior such CProcess and CCStructuralBase use multiple inheritance. This added behavior simplifies data entry for position information of all classes. As well, connection information for structural classes can be implicitly defined by their positions relative to the site and other objects.

4.3.3 CAPEBase

Provides basic object information, serialization, container membership, unique sequence number assignment, and attribute access interface.

4.3.4 CSimBase

This class will define basic virtual functions for simulation purposes. These functions provide various "user hooks" that allow the specialized classes to perform custom manipulation at the start and end of simulation and before and after each run. Virtual functions can be "overridden" by child classes in order to specialize.

4.3.5 CStructuralBase

This class adds connection management functionality. Classes like CRoadSegment and CIntersections, which constitute the structural elements of the model, derive from this class.

4.3.6 CProcess

A CProcess class defines basic functionality of a high level process such as ability to connect to interaction points and the base structures needed to define a simulation model. Child classes implement the specific simulation models.

4.3.7 CProject

Encapsulates all project level information and behavior. This includes the list of all processes, interaction point definitions, as well as any global resources that are available to all the processes.

4.4 Design Remarks

The classes provided above are used to construct the initial structure of the application. However, as the difficult task of coding a design progresses, a number of classes might still be added resulting in a slightly different class hierarchy.

5 CONCLUSION

Throughout the design process, every effort was made to enforce tight encapsulation and integration at all levels. The simulation design of each process is consistent and structured. Therefore, constructing a lower level representation such as CYCLONE (Halpin, 1990) is possible. At the project level, inter-process interaction through well defined points allows the tool to be easily extended. For example, if a compaction or a paving module is desired, another process is defined and linked without any changes being required of the remainder of the processes.

The design complexity in this paper is not very deep. The reason is that any more time spent on design will prove unproductive since any further design decisions are highly dependent on the implementation platform. The level of design discussed is general enough to be implemented using any object oriented framework with graphical user interface support. The design stage serves to examine all the various requirements and how they translate into modules and objects that will interact together. A good design will allow the application to gradually grow into a complex but well structured system.

The approach described in this paper was used in developing two special purpose simulation tools AP2_Earth (see Figure 10) and CRUISER. The first is a specialty tool for design and analysis of earth moving operations while the second is a specialty tool for analysis of aggregate production. Both tools are available via on ftp cem.civil.ualberta.ca/ public. They were described in the proceeding of the 1995 WSC (see AbouRizk et.al., 1995).

REFERENCES

AbouRizk, S. and Dozzi P. (1993). "Application of Computer Simulation in Resolving Construction Disputes". Journal of Construction Engineering and Management, ASCE. In press.

AbouRizk S. Shi, J., McCabe B. and Hajjar D.(1995) *"Automating the process of Building Simulation Models"*. Winter Simulation Conference, December 1995 Washington D.C.

Goldberg, A. and Robson, D. (1983). SmallTalk-80: The language and Its Implementation. Addison-Wesley Publishing Company, Reading, MA.

Halpin, D. W. (1977) "CYCLONE: Method for Modeling of Job Site Processes." *Journal of the Construction Division*, ASCE, 103(3), 489-499.

Halpin, D., Woodhead, R., *(1976). Design of Construction Process Operations,* John Wiley and Sons, Inc., New York, NY, 1976.

Liu L.Y., and Ioannou P.G., (1992). "Graphical Object-Oriented Simulation System for Construction Oriented Discrete-Event Simulation System." *Proceedings of the 1992 Winter Simulation Conference 12*85-1291.

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

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

Miller A., Glenda G. and Bird C. (1991). "Simulation and Analysis of Computer Systems With Multiple Supercomputers", SCS Multi-Conference, New Orleans, Apr 1-5, 1991.

Odeh, A. Tommelein I, and Carr R. (1992). "Knowledge-Based Simulation of Construction of Construction Plans". Proceedings of the Eighth

Conference on Computing in Civil Engineering, ASCE, Dallas, Texas. 1042-1049
Zeigler, B.P. (1987). Hierarchical, Modular Discrete-Event Modeling in an Object-Oriented Environment, Simulation, November 1987, 219-230.

ACKNOWLEDGEMENT

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant CRD 166 483.

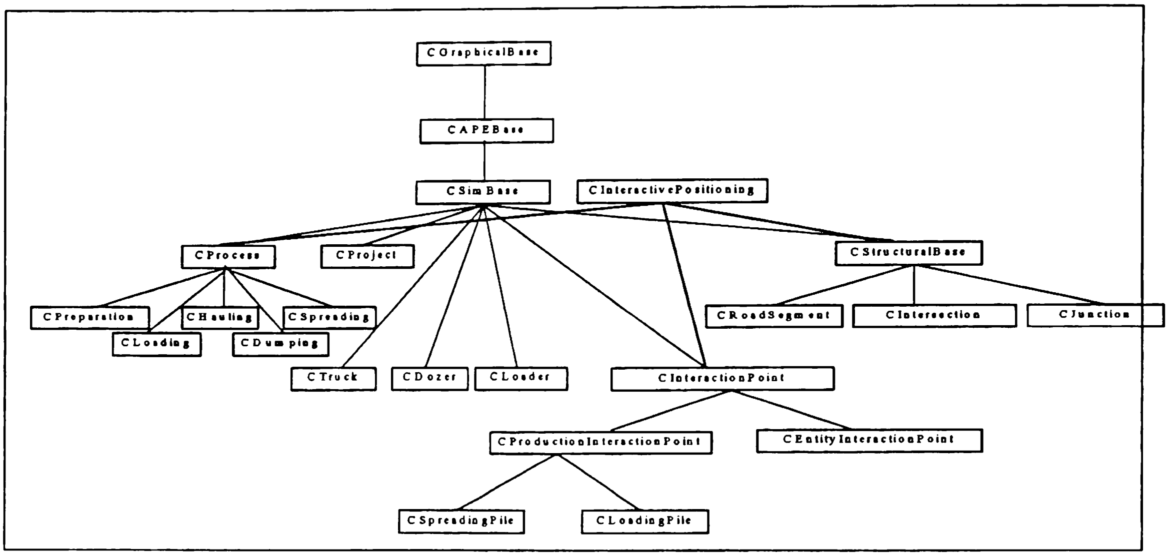

Figure 7: Class Hierarchy

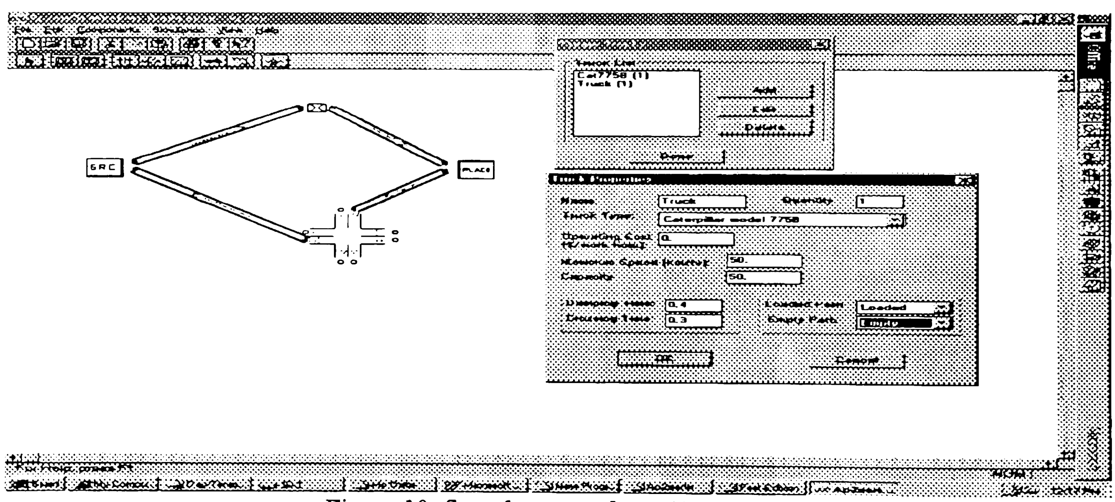

Figure 10: Sample screen from AP2_Earth

AUTHOR BIOGRAPHIES

DANY HAJJAR is a provisional 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 interest are focused on applying automation and object oriented methodologies to the management and control of construction projects..

SIMAAN M. 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. He received his Ph.D. degree from Purdue University in 1990. His research interests are mainly focused on the application of computer methods and simulation techniques to the management of construction projects.