OBJECT ORIENTED SIMULATION OF EARTHMOVING OPERATIONS

Amr A. Oloufa, Ph.D., P.E.
University of Hawaii, Department of Civil Engineering
2540 Dole Street, Holmes 383, Honolulu HI 96822, USA

ABSTRACT

The objective of this paper is to identify and study object oriented constructs exhibiting feedback mechanisms required for the operational simulation of construction projects. To illustrate the concepts involved, we chose a demonstration in earthmoving operations.

Contrary to manufacturing operations, the "factory floor", i.e. the construction site, has a profound impact on the performance of operational resources. Any meaningful representation of construction operations should therefore include, as an integral part, the construction site. The proposed paper emphasizes an integrated approach to the solution of this problem. In our approach, the various components exchange information during simulation. This guarantees that performance parameters are constantly updated as the simulation continues. By updating performance parameters as the simulation proceeds, we are taking into consideration the impact of the construction site on the productivity of equipment, an element that has considerable ramifications on system's performance.

Through the creation of a library of object oriented constructs representing construction resources, the construction team can create the simulation model by specifying its component objects. The utilization of object oriented elements in the model facilitates its creation, gives the simulation model the ability to update its instance variables through the simulation and enhances the comprehensibility of the simulation model. The behavior of the model through simulation is displayed with the aid of animated computer graphics.

SIMULATION IN CONSTRUCTION

Construction operations, similar to production operations, can be broken down to a series of work tasks having static or stochastic durations. In contrast to production operations, construction operations are usually repetitive in nature, mostly labor intensive and carried out in open environments making them susceptible to factors such as weather conditions that are hard to predict or control.

We have selected earthmoving operations to highlight specific problems that are unique in the operational simulation of construction activities. Earthwork operations are highly dependent on the typographic changes that occur during this type of construction and afford an excellent model for explaining advantages of this approach.

Lux in [1] lists the following factors that influence equipment choice in earthmoving operations: (1) Kind of material: Abrasiveness, Weight, Moisture Content and Soil Condition. (2) Location and condition of cut and fill (Excavation and Backfill). (3) Length of haul road. (4) Grades and rolling resistance of road. (5) Condition of haul road. (6) Volume of material moved. (7) Job duration. (8) Job cost : labor and fuel. (9) Other considerations : Traffic Flow, Noise Level, and Working Hours.

WORK TO DATE

Simulation analyses employed in earthmoving applications resembled the flowchart shown in fig.(1).

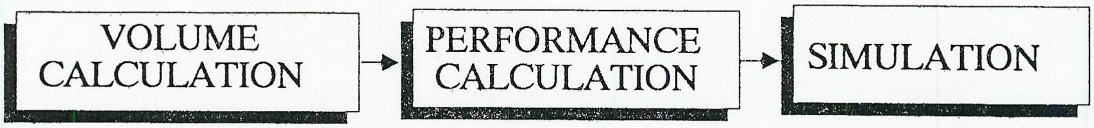

Fig.(1) Flowchart of Analyses Employed

Following the early work at Stanford University by Fondahl, Gaarslev and Teicholz [3,4,5], Halpin [6] developed CYCLONE (CYClic Operation NEtwork) using a graphical modeling format which is an adaptation of GERTS IIIQ, a member of the GERTS family of programs developed by Pritsker and Associates [7]. GERT (Graphical Evaluation and Review Technique) is a method for analyzing networks which have stochastic and logical properties. At Stanford University, CYCLONE modeling capability was extended and the program has undergone several modifications including porting it to an interactive microcomputer environment and utilizing computer graphics [8,9].

Shown in fig.(2) from [7], is a CYCLONE network for an earthmoving operation. When a loader, earth and a truck are available, the loading can begin. After the truck is loaded, the loader returns to an idle state and the truck travels to the dump site. When the dumping activity is completed, the truck returns and joins the queue for the loader.

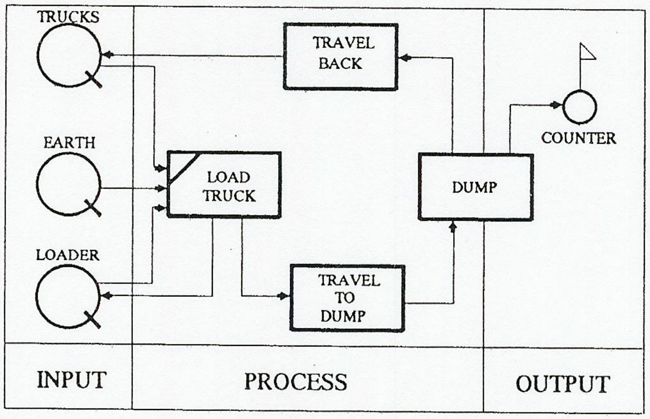

Fig.(2) CYCLONE Network

Woodhead and O'Brien at the University of New South Wales in Sydney, Australia have concentrated on a time-lapse analysis system [10,11]. Bjornsson at Chalmers Institute in Sweden has experimented with micro-computer based video-tape acquisition. Carr at the University of Michigan has worked with techniques to deal with broader classes of uncertainty in the operating environment [12,13]. Kavanagh at the University of Missouri-Rolla has proposed a computer model for repetitive construction operations by using a personal computer to prepare the code for use by a remote mainframe running GPSS [14]. Ward [1], presented a computer-aided technique that is being developed for the planning, design, estimating and scheduling of horizontal construction facilities required for the support of military installations.

Because of the nature of earthmoving operations, Oloufa [16] proposed an integrated simulation approach as shown in fig.(3). Fig.(3) is a "Utopian" view of the important elements and their interaction. New elevations or soil types are considered since they will affect performance parameters such as speed. The ability to use an animated graphics representation for the construction site as shown in fig.(4) has some very important and desireable features. First, as a communications tool, the model is comprehensible to the construction project team since it employs familiar building elements such as trucks, loaders and construction site contours. Also performance inconsistencies can be easily detected by viewing the animated representation of the operation.

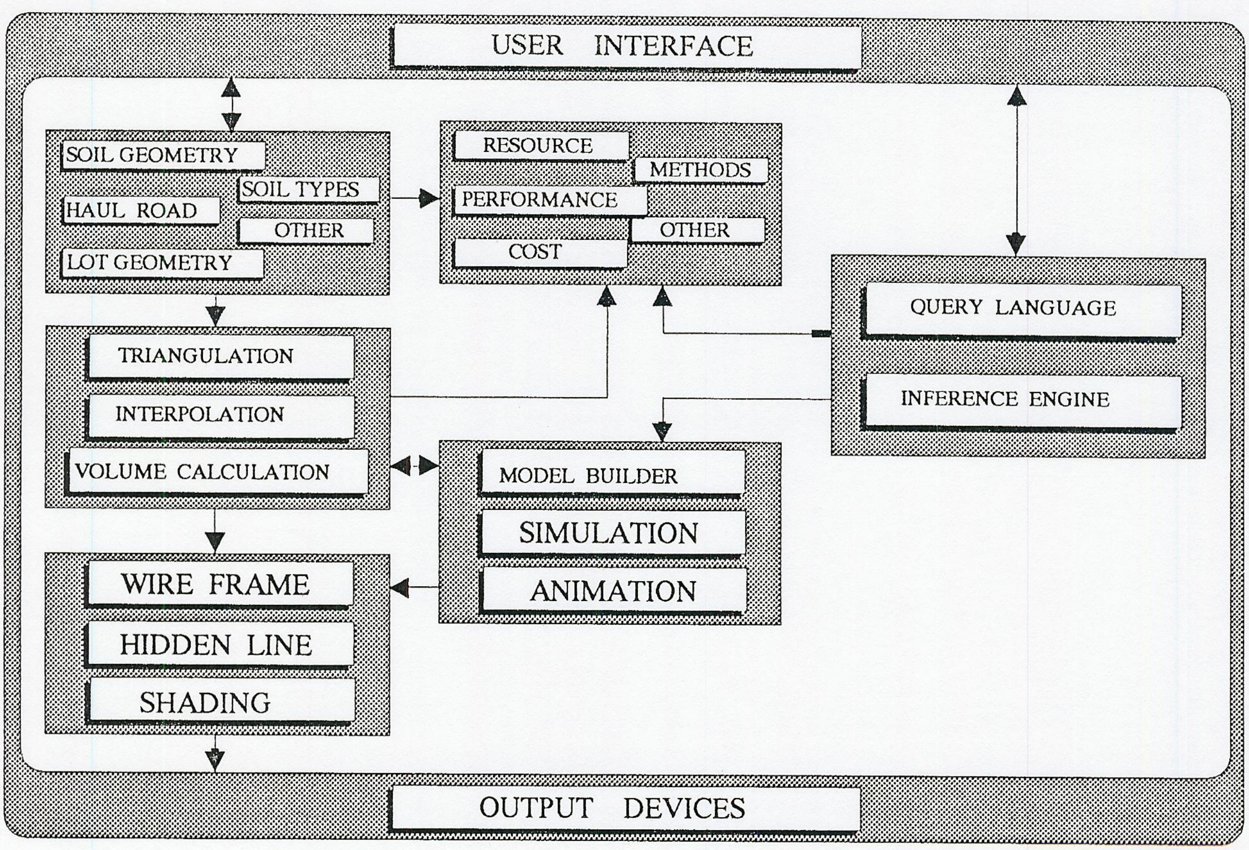

Fig.(3) The Utopian View

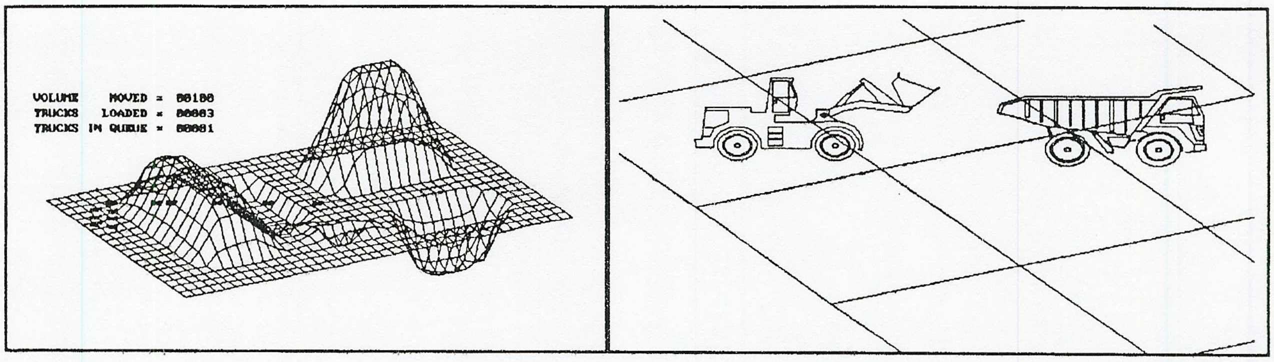

Fig.(4) Animated Simulation of Earthmoving Operations

WEAKNESSES OF CURRENT APPLICATIONS

Communication is a cornerstone for a meaningful simulation and it cannot be achieved unless the model and its output are completely comprehensible to the end-users. It is desireable to develop a modeling methodology that is both capable of *accurate representation and easy to build models with.* It is also important to view the construction site *during* the different stages of simulation since changes in soil type or topography may lead to a better insight into the selection of different equipment or construction methods. Most current modeling methodologies fail to consider the fact that the construction site ("factory floor") is always changing with time affecting performance and durations of the various work tasks.

Current simulation languages that have an animation capability like CINEMA, CADMOTION, SIMSCRIPT and SLAMSYSTEM do not have the ability of "updating" the simulation background, or in our application, the construction site. Also, their animation capability was added to the simulation program in a later date leading to an unnatural modeling environment that worsens, even further, the correspondence between the model and its real world counterpart. There is a definite need for a simulation paradigm and language that has animation as a "built in" capability rather than an "add-on" feature.

THE "OBJECT-ORIENTED" APPROACH

The object oriented view is that the system is composed of interacting physical objects. Objects may be trucks and scrapers in an earthmoving operation or a crane in a high rise building. These objects are typically the central focus of the simulation studies. The simulation problem is one of finding convenient means of modeling these objects in an effort to simulate their behavior.

The modeling perspective employed in simulation is directly related to the modeling tools available for the simulation language being used. During the 1950's, simulation was exclusively done using general purpose programming languages. In the 1960's, GASP, SIMULA and SIMSCRIPT were among the early general purpose simulation languages developed. These languages contained specific simulation constructs and procedures to aid the programmer in developing the model.

Later in the 1960's, a different approach was proposed with the introduction of GPSS. The novelty of GPSS was its emphasis on a modeling structure that hides from the user the mechanics of the simulation. GPSS was built upon a predefined class of active entities called "transactions" which flowed through a flowchart of selected operations; similar to the programming flow charts of procedural languages.

In the 1970's and 80's, several languages like SIMSCRIPT, GPSS/H, SLAM and SIMAN, were introduced and/or modified. These languages have tried to satisfy the dual goals of generality (found in general programming languages) and convenience (provided by specialized simulation languages) by providing a set of predefined objects and concepts for direct modeling.

SIMULATION WITH OBJECTS

In describing a system, we define its components and how these components interact together. We also declare the valid operations these components engage in. We also specify how this engagement affects their states before, during and after these operations. If we consider an earthmoving operation, we define the trucks, scrapers, bulldozers and other machinery involved. We also define their interaction and the likely outcome/s of their association. The states of these equipment are described before they engage in their respective operations, during, and after the operations are completed.

In the process of system modeling, the components of the system, how they interact and the possible outcomes of their interaction are identified. The next step is the mapping of these components to the respective terminology in the modeling environment. We are faced with a situation that what used to be comprehensible physical components have now to be represented in terms of transactions, queues, resources, attributes, etc. in a network-oriented simulation language.

Readability of simulation software is the property of facilitating the comparison between the physical system and its representative models. Readability is concerned with the separation of the components describing the model from the information required to gather statistics or display the behavior of the system. A major constituent for the success of any modeling environment is the readability aspect. The ability to understand the representation of the simulation model should be reserved to the professional programmer. It should also be extended to the end user who is usually the specialist in the application field and the person who is mostly familiar with the problem.

Object oriented approaches are an attempt to bridge the gap between the physical system and its computer representation. It is a methodology that minimizes the need to "understand and learn" a model that presumably mimics a physical system whose components may already be well understood. In an object oriented environment, "objects" or "encapsulated data structures" are used to represent the various components in our system.

Each object represents a physical component of the system being modeled. The set of objects that represent the same kind of system component is called a class. A class definition is used to define a particular abstract data type. An abstract data type specifies its own operations in addition to its own characteristics.

A class has individual objects called instances. A class describes the form of its instances and how they carry out their operations. Objects communicate with other objects using messages. A message is a request that an object carry out one of its operations. A message specifies the operation desired but not how it is done. The receiving object determines how to carry out the operation requested by the message. The set of messages to which an object can respond is called its interface with the rest of the system. The object's operations can be invoked only by the messages its receives via its interface. When an object receives a message, an operation is invoked. This operation is controlled by a specific method stored within the object's definition. An object's response to the messages it receives is governed by its methods. Each method stores information on how to respond to a specific message received by the object. Each object may have public and private methods. Public methods provide the "outside" interface to the class whereas private methods provide the interface between the objects of the same class.

One of the most powerful features of object oriented languages is inheritance through hierarchial system description. The principle feature of class hierarchy is that any class inherits all the properties from its superclass. Each subclass has the properties of its super class and may also have properties that are unique to it. Some object oriented languages also permit the ability to inherit properties and methods from other classes. Inheritance facilitates programming since new classes need only be specified by their difference from an existing class rather than having to be defined from scratch.

IMPLEMENTATION OF THE OBJECT ORIENTED APPROACH

The Smalltalk object-oriented language supports discrete event simulation by providing support for simulation classes. These classes include Simulation, SimulationObject, DelayedEvent, WaitingSimulationObject, Resource, ResourceProvider and ResourceCoordinator. Smalltalk may be used for simulation by defining models based on a client-server paradigm. However, Smalltalk lacks support for time queues and other queueing with process control, limiting its applicability in general simulation modeling.

SimTalk, an extension of Smalltalk, developed by the Tektronix Corp. [18] defines a simulation control class which maintains the time queue, the simulated clock and contains other classes for the generation of random numbers, probability distributions, statistics gathering and animation functions. In our implementation, SimTalk is used to model the earthmoving operation.

In developing our model, we start by defining the project site in terms of interacting objects. Examples of objects are equipment, labor crews and the construction elements such as contour lines, soil type, haul road profile etc.

Through the progress of the simulation, the soil and haul road "objects" can modify their attributes (i.e. slopes, soil types, rolling resistance) based on the amounts of cut and fill. This progress is communicated via messages to the other simulation classes and objects such as equipment. In turn, the equipment classes modify their performance depending on the "messages" received from the soil and haul road objects and the simulation continues...

Using Smalltalk and SimTalk, we can develop a modeling environment that addresses most of the weaknesses of current applications described before.

CONCLUSION

Contrary to manufacturing operations, the "factory floor", i.e. the construction site, has a profound impact on the performance of operational resources. Any meaningful representation of the construction operations should therefore include, as an integral part, the construction site.

Object oriented approaches introduce a superior modeling philosophy by representing simulation entities in a format that closely resembles their real life origins. This approach reduces the cognitive gap between the system and its representation. It focuses the developer's attention to the actual "object" rather than the "processes" in the system. Objects provide both data abstraction and information hiding leading to a modular approach in modeling. Each object encapsulates its "capabilities" within its definition yielding a natural decomposition of the system. Another major benefit is the ease of modifying the simulation model. Construction operations involve unexpected situations that require a quick response. The benefits and ease of developing a model using modular constructs that embody their capabilities (i.e. methods) are obvious. Also use of the object oriented approach leads to a reduction in the resulting code [25]. Reducing code size improves the readability of the simulation model and makes it easier to comprehend.

A major limitation in the use of object oriented approaches is the computational overhead that is required. It is expected however that this weakness will diminish with the emergence of more powerful machines with parallel processing.

ACKNOWLEDGEMENTS

The author would like to thank the University of Hawaii Research Council for their support. Also special thanks to Rick Mootry and Verna Knapp at Tektronix Labs in Beaverton, OR.

REFERENCES

[1]- Oberlander, G.: *"Earth Moving & Heavy Equipment"*, Proceedings of the Conference of the Construction Division of ASCE,Tempe, Arizona, 1986.

[2]- Paulson, B. C., Weng, T. C. and Soo, C. L.: *"Simulating Construction Operations by Microcomputer"*, Construction Research, ASCE, 1985.

[3]- Fondahl, J. W.: *"Photographic Analysis for Construction Operations"*, Journal of the Construction Division, ASCE, Vol.86, No. C02, May 1960.

[4]- Gaarslev, A.: *"Stochastic Models to Estimate the Production of Material Handling Systems in the Construction Industry"*, Technical Report No. III, The Construction Institute, Stanford University, Aug. 1969.

[5]- Teicholz, P.: *"A Simulation Approach to the Selection of Construction Equipment"*, Technical Report #26, The Construction Institute, Stanford University, 1963.

[6]- Pritsker, A. and Burgess, R.: *"The GERT Simulation Progress: GERTS III, GERTS IIIQ, GERTS IIIC and GERTS IIIR"*, NASA / ERC Contract NAS-12-2113, Departmental Report, Civil Engineering Department, Virginia Polytechnic Institute, 1970.

[7]- Halpin D. W. and Woodhead, R. W.: *"Design of Construction and Process Operations"*, John Wiley & Sons, New York, 1976.

[8]- Kalk, A. and Douglas, S.: *"Insight: Interactive Simulation of Construction Operations using Graphical Techniques"*, Technical Report No. 283, The Construction Institute, Dept. of Civil Engineering, Stanford University, 1980

[9]- Paulson, B. C., Douglas, S. A., Kalk, A., Touran A. and Victor G.: *"Simulation and Analysis of Construction Operations"*, Journal of the Construction Division, ASCE. Vol. 109, Aug. 1983.

[10]- Dorman, W. F., J. R. Knott and R.W. Woodhead: *"Cyclone Time-Lapse Analysis System, Introductory User's Manual"*, School of Civil Engineering, University of New South Wales, Kensington, New South Wales, Australia, December, 1980.

[11]- O'Brien, J. B., J. L. Knott and R. W. Woodhead: *"A System for Fully Automated Capture and Analysis of Field Performance Data"*, CIB W-65, Proceedings of the 4th International Symposium on Organization and Management of Construction, Volume II, Waterloo Ontario, Canada, 1984.

[12]- Carr, R. I.: *"Simulation of Construction Project Duration"*, Journal of the Construction Division, ASCE, Vol. 105,No. C02, Proc. Paper 14607, June 1979.

[13]- Carr, R. and Meyer, W.: *"Planning Construction of Repetitive Building Units"*, Journal of the Construction Division, ASCE, Vol. 100, No. C03, Proc. Paper 10812, Sep., 1974.

[14]- Kavanagh, D.: *"SIREN: A Repetitive Construction Simulation Model"*, Journal of Engineering and Management, ASCE, Vol.III, Sep. 1985.

[15]- Bernold, L. E.: *"Productivity transients in Construction Processes"*, Ph.D. Thesis, Georgia Institute of Technology, 1985.

[16]- Oloufa. A. A.: *"A Framework for the Operational Simulation of Construction Projects"*, Department of Civil Engineering, University of California, Berkeley, Ph.D. Dissertation, May 1988.

[17]- Knapp, V.: *"The Smalltalk Simulation Environment"*, Proceedings of the 1986 Winter Simulation Conference, Washington, D.C.

[18]- Knapp, V.: *"The Smalltalk Simulation Environment, Part II"*, Proceedings of the 1987 Winter Simulation Conference, Atlanta, GA.

[19]- Roberts, S. D. and Heim, J.: *"A perspective on Object Oriented Simulation"*, Proceedings of the 1988 Winter Simulation Conference.

[20]- Bezivin, J.: *"Timelock: A Concurrent Simulation Technique and Its Description in Smalltalk-80"*, Proceedings of the 1987 Winter Simulation Conference, Atlanta, GA.

[21]- Rothenberg, J.: *"Object-Oriented Simulation: Where Do We Go From Here?"*, Proceedings of the 1986 Winter Simulation Conference, Washington, D.C.

[22]- Cammarata, S., Gates, B. and Rothenberg, J.: *"Dependencies and Graphical Interfaces In Object Oriented Simulation Languages"*, Proceedings of the 1987 Winter Simulation Conference, Atlanta, GA.

[23]- Bezivin, J.: *"Design And Implementation Issues In Object Oriented Simulation"*, Simuletter, 1988.

[24]- Kim, T. and Zeigler, B.: *"The Class Kernel-Models Scheme: A Hypercube Architecture Example"*, Simuletter, 1988.

[25]- Cox, B. J., *"Object Oriented Programming: An Evolutionary Approach"*, Addison-Wesley, Reading, MA, 1987.