A KNOWLEDGE-BASED SIMULATION SYSTEM FOR CONSTRUCTION PROCESS PLANNING

David Y. Chang
Texas A&M University

Md. Saiful Hoque
Gulf Interstate Engineering

ABSTRACT

Discrete event simulation has not yet been accepted by the construction industry as a tool for process, or field level operations, planning. This may be attributed to the fact that its use requires considerable knowledge of both the construction process and simulation technique. To overcome these impediments, a new knowledge-based simulation framework has been developed. This framework uses the object-oriented knowledge representation to separate specific construction and simulation knowledge from end-user applications, and thereby simplify and enhance the design and uses of simulation models. This paper identifies the need for a construction process planning tool; reviews the evolution of process simulation tools; and presents the knowledge-based simulation framework for construction process planning. System implementation and future extensions are also discussed.

INTRODUCTION

The complexity and enormity of modern construction projects dictate the use of sophisticated planning tools and technologies that are capable of efficient allocation and management of available resources. In construction four levels of planning hierarchy exist: corporate, project, process and operation planning. Corporate and project level planning is concerned with setting company long-term and short-term goals, project portfolio and coordination, contract administration, project cost and schedule. Whereas process and operation level planning is concerned with selection of construction methods, commitment of resources, and daily progress.

The most common planning tool in construction is the critical path method (CPM) networking technique. The CPM technique helps managers to allocate, schedule and control resources at the corporate and project levels. It is not, however, appropriate for planning construction processes and operations. This is because the CPM technique is limited to cost/time relationships among project activities and it fails to account for actual interaction among resources at the process level. As a consequence, field managers must use other techniques, coupled with their knowledge, experience, and judgement, for construction process planning.

In the following sections, this paper identifies the need for a construction process planning tool; reviews the evolution of process simulation tools; and presents an ideal knowledge-based simulation framework for construction process planning.

System implementation and future extensions are also discussed.

NEED FOR A CONSTRUCTION PROCESS PLANNING TOOL

A construction process can be defined as a collection of basic field operations that must be executed in a certain order to complete a specific project work item. One example is an earthmoving process which involves the following basic operations: loading, hauling, dumping, and returning.

At the process level, planning involves selection of construction methods and allocation of resources with the aim to optimize process production. The selection of construction methods and resources depends on numerous factors such as site conditions, engineering properties, work quantities, resource availability, crew skills and time constraint. A good construction process planning tool, therefore, should be able to consider individual factors and their interactions in developing a plan.

If these factors and their interactions are simple enough, it may be possible to use mathematical or graphical methods to derive and compare different plans. Example methods are equipment balancing, line-of-balance, and queueing models [4,7]. These methods use point estimates or specific distributions to calculate operation times, and use correction factors to account for variations in process conditions. Unfortunately, most construction processes involves interactions that are too complex to evaluate mathematically and graphically. As a consequence, these methods do not yield realistic results to reflect the behavior of complex construction processes. There is a need for a construction process planning tool.

Discrete event simulation has been used by the manufacturing industry as a planning tool for analyzing complex manufacturing systems. In a simulation, a model is built to represent a real-world situation, a computer is used to evaluate the model numerically over a certain time period and data are gathered to estimate the true behavior of the model. A number of general purpose simulation languages are used in modeling manufacturing processes. The well-known ones are GPSS, SIMSCRIPT, SLAM-II, and SIMAN.

Although the simulation technique is a proven process planning tool for manufacturing, it has not been adopted by the construction industry. The lack of technology transfer can be traced to the following factors:

Construction processes are unique and short-term. A manufacturing process is generally stable and long-term. One example is a circuit board assembly line, once set up it will be operated in the same way for a long period of time. Therefore, the potential cost saving resulting from good planning has motivated management to invest in simulations. This is not the same in field construction, because each project is unique, with different conditions, and short-term.

Construction processes are intermittent. Typical manufacturing processes involve various stationary work stations (servers), work-in-progress is sent to these homogeneous servers in a particular sequence. Most field construction processes, on the contrary, involve sending heterogeneous servers to stationary work-in-progress.

Consequently, simulation languages developed for manufacturing cannot be easily adapted to model construction processes [5,8].

Construction processes involve complex resource interactions. As indicated in the last paragraph, construction processes are driven by the properties and interactions of resources. These interactions cannot be easily modeled by existing simulation languages.

CONSTRUCTION PROCESS SIMULATION

A number of simulation systems have been developed to model construction processes. They range from simple deterministic systems, such as VEHSIM, to sophisticated ones, such as CYCLONE. VEHSIM [3] is a deterministic simulation system developed by an equipment manufacturer to compare production and cost of haulers under different hauling conditions.

CYCLONE, an acronym for CYCLic Operations NEtwork system, is the pioneer of the more advanced construction simulation system. It uses a set of simple graphic elements to model resource flows and requirements typical in construction processes. The system is fully defined in [8].

The major CYCLONE modeling elements are Arc, Queue, Combi, and Normal. An Arc is an arrow, depicting the direction of resource flow. A Queue is a circle with a slash, representing the waiting line of resources. A Combi is a square node with a slash at the top left hand corner, representing an operation that requires resources from its preceding Queues. That is, a Combi must follow Queues, and Queues must precede Combis. A Normal is a square node, representing an operation that can be initiated once its preceding operation is finished.

Figure 1 is a CYCLONE model of a simplified manual earthmoving process. Resources involved are loader, wheelbarrow, and pusher. Operations, resource required, and resource flows are depicted by a combination of the four modeling elements.
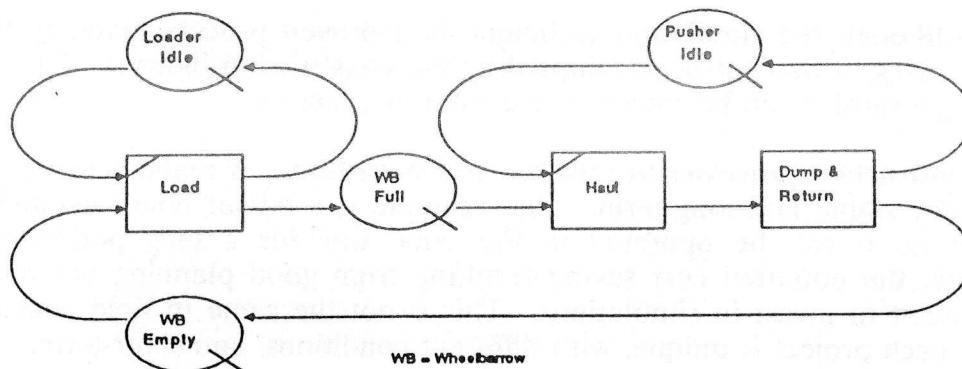

Figure 1 CYCLONE Model of a Simplified Manual Earthmoving Process

The major strengths of the original CYCLONE are: (1) a simulation model can be easily developed and communicated; (2) basic resource requirements and

flows can be easily formulated; and (3) different process configuration can be studied to help balance resources. Its major weaknesses in modeling construction processes are: (1) the lack of resource representations; and (2) the absence of adequate control modeling structure.

CYCLONE assumes that resources flowing through the same Arc are identical and therefore will have the same behavior. If non-identical resources are present, as is frequently so in construction, the model must be exploded in order to model the variations. Figure 2 shows the new model of the example process if two non-identical pushers and two different capacity wheel barrows are employed.

Resource flows in CYCLONE must be modeled by a combination of modeling elements; no explicit control structure is available. Therefore, control strategies such as routing resources based on selected Queue lengths, are difficult to model.

Refinements and expansions of the CYCLONE system have been developed. Examples are MicroCYCLONE [2] and automated data collection system [11]. Since all these systems are based on the original CYCLONE modeling concepts, they inherit the intrinsic CYCLONE system weaknesses.

RESQUE [4,5], an acronym for RESource based QUEueing network simulation system, is another significant extension of CYCLONE. It adds an overlay structure to the CYCLONE-like process models. In the overlay structure, resource properties, operations and their interactions can be defined by using the RESQUE's Process Description Language (PDL). Therefore, complex resource interactions can be defined without complicating the graphical model. A detailed description of RESQUE can be found in [4,5]. Figure 3 shows the RESQUE model for the same example process with non-identical resources.

The strengths of RESQUE are: (1) non-identical resource units and their interactions can be easily defined without complicating the model; and (2) the rich modeling construct and built-in inferencing power allow the formulation and evaluation of different control strategies. Its major weaknesses are: (1) process detailing with PDL statements is batch oriented, which can be tedious and error prone; and (2) resource interaction and control strategy definitions are embedded in the batch definition, which cannot be easily modularized and reused.

These construction process simulation tools have been successfully used in classroom teaching; however, the construction industry has been slow in adopting them. The major impediment is probably the requirement of extensive knowledge in the areas of probability, statistics, design of experiments, computer languages and simulation modeling. Furthermore, extensive knowledge of construction processes is also required in order to formulate feasible alternatives.

An ideal process planning tool for the construction industry, therefore, should have these abilities: (1) provide assistance with experiment design and analysis; (2) provide a means to capture and reuse construction process knowledge; and (3) provide a graphics-based user interface that allows direct manipulation of modeling elements and process knowledge at a higher level.
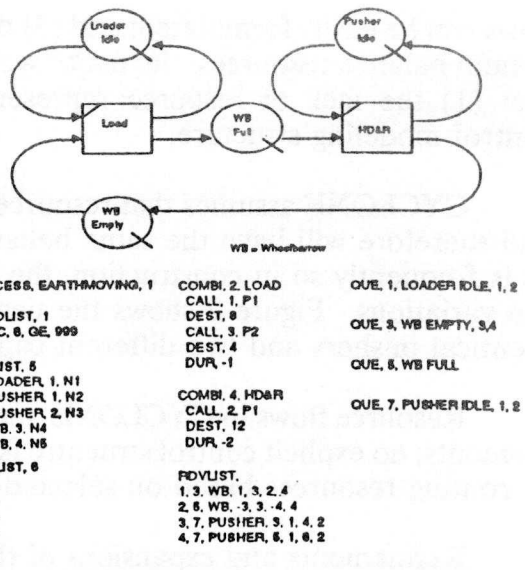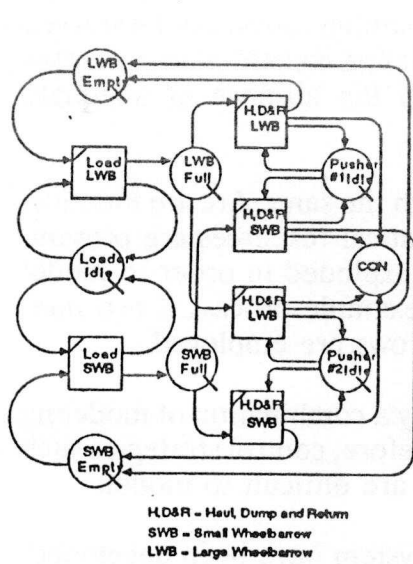
H.D&R = Haul, Dump and Return
SWB = Small Wheelbarrow
LWB = Large Wheelbarrow

Figure 2. CYCLONE Model of the Example Process

WB = Wheelbarrow

PROCESS, EARTHMOVING, 1	COMBI, 2, LOAD	QUE, 1, LOADER IDLE, 1, 2
	CALL, 1, P1	
CONDLIST, 1	DEST, 6	QUE, 3, WB EMPTY, 3,4
1, CC, 6, GE, 999	CALL, 3, P2	
	DEST, 4	QUE, 5, WB FULL
RESLIST, 5	DUR, -1	
1, LOADER, 1, N1		QUE, 7, PUSHER IDLE, 1, 2
2, PUSHER, 1, N2	COMBI, 4, HD&R	
2, PUSHER, 2, N3	CALL, 2, P1	
3, WB, 3, N4	DEST, 12	
3, WB, 4, N5	DUR, -2	
DURLIST, 6	RDVLIST.	
1, 2	1, 3, WB, 1, 3, 2, 4	
2, 2	2, 5, WB, -3, 3, -4, 4	
3, 2	3, 7, PUSHER, 3, 1, 4, 2	
4, 2	4, 7, PUSHER, 5, 1, 6, 2	
5, 2		
6, 2		

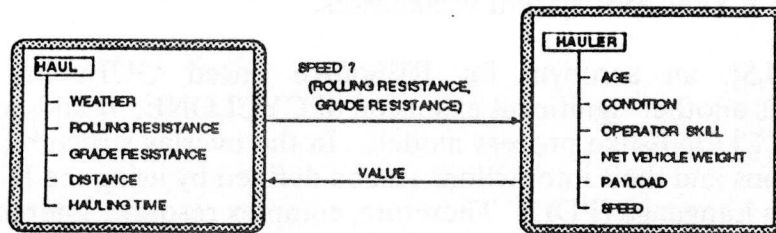**Figure 3. RESQUE Model of the Example Process**

Figure 4. Object Abstraction and Message Sending
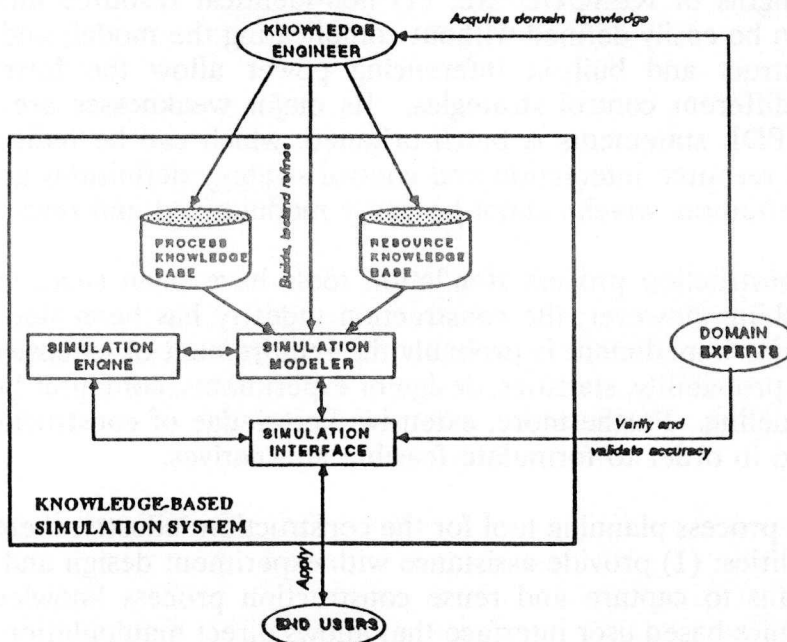

Figure 5. Knowledge-Based Simulation Framework

352

KNOWLEDGE-BASED CONSTRUCTION SIMULATION FRAMEWORK

To achieve these goals, a knowledge-based simulation framework has been developed by the authors. The framework will allow the construction industry to capture and reuse the most important asset - construction process knowledge - to accurately simulate and plan one crucial part of its business - field operations.

The two major concepts of this framework are **object-oriented knowledge representation** and **multi-level model construction.**

The **Object-oriented knowledge representation** is a programming paradigm that has its root in both the object-oriented programming language and the frame-based knowledge representation [1,6]. This paradigm is represented by the following features: **data abstraction, message sending, inheritance,** and **late binding.**

In this paradigm, each object is an **abstraction** or encapsulated unit with its own behavior, representing a real world counterpart. Objects can only interact by way of **sending messages.** The behavior of an encapsulated object can be static or dynamic. The static behavior is represented by constant attributes and the dynamic behavior is represented by methods. A method is a procedure used by an object to respond to a particular message. These features provide a direct way to model real-life objects and their interactions.

Figure 4 shows one example where two encapsulated objects, the "haul" operation and one "hauler", interact through sending messages. To get the haul speed, the haul operation must send a message "speed" to the hauler with information about grade and rolling resistance. This message will activate the hauler "speed" method to calculate and return the haul speed based on information sent and its own "net vehicle weight" and "payload" attributes.

Objects in this paradigm are defined in a taxonomic hierarchy consisting of class, subclass, and instance. A class is an object that defines the generic behavior of a group of objects. A subclass is a more specialized class. An instance is an active object that **inherits** behavior from its higher classes and/or subclasses. additionally, an instance can have its own behavior defined. This feature allows the incremental definition of objects and the sharing of data storage.

This paradigm will not **bind** a method with its pertinent attributes until the latest point, i.e., at run time. This ensures the true behavior will be reflected through object interactions.

The object-oriented paradigm provides a means to modularize specific construction process knowledge. These knowledge modules are independent but interrelated objects; they can be linked together through proper interfaces to formulate simulation models.

The **multi-level model construction concept,** as shown in Figure 5, divides the development of a simulation model into two levels: expert and user. At the expert

353

level, knowledge modules are developed by the domain expert with the assistance of the knowledge engineer. There are two types of knowledge modules: construction process and construction resource. A construction process module is an ordered list of basic operations for a generic process. A construction resource module is an object describing the behavior of a particular resource unit. These modules are respectively stored in the process and resource knowledge bases.

At the user level, the system will assist the user to identify process and select proper construction resources based on specific process requirements and conditions. The selected process and resources will be retrieved from the knowledge bases and presented as graphic symbols similar to the ones used in CYCLONE and RESQUE. The user can directly manipulate these symbols without the knowledge of their internal representation. The system will maintain the model integrity and consistency.

This knowledge-based simulation framework enhances as well as simplifies construction process simulations. Specific construction process knowledge can be captured and used to guide the user. To the user, formulation and evaluation of different process plans will be as easy as playing a computer game. This is what simulation is all about.

SYSTEM IMPLEMENTATION AND FUTURE EXTENSIONS

Research is in progress to implement this knowledge-based simulation framework for the earthmoving process. The research team has selected the Simkit simulation package, an extension of the KEE software, as the implementation tool [9,10].

KEE is a general purpose hybrid expert system shell that incorporates frame-based, object-oriented, and rule-based knowledge representations. Simkit is a simulation shell built on top of KEE; it has a kernel made up of object-oriented simulation routines such as simulation clock, event calendar, random number generator, simulator, and data collector. Moreover, it has an interactive icon-based editor that can tailor the desired direct manipulation end-user environment.

The resulting system will provide a foundation for significant extensions, including: (1) develop applications for other construction processes; (2) build a simulation analyzer that will help analyze simulation results, make appropriate recommendations, or diagnoses; (3) build an interface that will read CAD data as process description, eliminating the need to re-extract information; (4) incorporate specific knowledge that will enable the system to become a "proving ground" for construction robots; and (5) develop a goal-driven simulation inference engine, namely the user can define the process of interest and specific production goals, and let the system search for answers.

CONCLUSION

A knowledge-based simulation framework is presented. This framework provides a seamless environment where specific construction knowledge can be

integrated with simulation to simplify as well as enhance process planning. Research is in progress to implement the modeling framework. As we observe the latest developments in the areas of computer aided design, information management, project and process planning, it is inevitable to notice the trend and importance of integrating information flows in the emerging electronic work environment, and to conclude that profound changes will result from the advances in computer hardware and the development of an unifying information representational paradigm.

ACKNOWLEDGEMENT

REFERENCES

1. Barr, A., and Feigenbaum, E.A. (Ed.), The Handbook of Artificial Intelligence, Volume 1, William Kaufmann, Inc., Los Altos, CA, 1981.

2. Bernold, L., and Halpin, D.W., "Advanced Microcomputer Simulation for Construction Managers," Computing in Civil Engineering, Proceedings of the Fourth Conference, American Society of Civil Engineers, October 1986, pp. 762-784.

3. Caterpillar, Vehsim 2.0 User's Guide, Caterpillar Inc., Peoria, IL, 1987.

4. Chang, D.Y., "A Resource Based Simulation Language for Construction Process Planning," Ph.D. Dissertation, Civil Engineering Department, University of Michigan, Ann Arbor, 1986.

5. Chang, D.Y., and Carr, R.I., "RESQUE: A Resource Oriented Simulation System for Multiple Resource Constrained Processes," Proceedings of the 19th Annual Project Management Institute Seminar/Symposium, Milwaukee, WI, October 1987.

6. Cox, B.J., Object Oriented Programming, Addison-Wesley Publishing Company, Reading, MA, 1986.

7. Halpin, D.W., and Woodhead, R.w., Design of Construction and Process Operations, John Wiley and Sons, Inc., New York, NY, 1976.

8. Hoque, M.S., "A Knowledge-Based Object-Oriented Simulation Framework for Construction Process Planning," Master of Science Thesis, Civil Engineering Department, Texas A&M University, December 1988.

9. IntelliCorp, The Simkit System Knowledge-Based Simulation Tools in KEE, User's Manual, IntelliCorp, Inc., Menlo Park, CA, 1987.

10. IntelliCorp, KEE Software Development System User's Manual, IntelliCorp, Inc., Menlo Park, CA, 1986.

11. Paulson, B.C., Chan, W.T., and Koo, C.C., "Construction Operations Simulation by Microcomputer," Journal of Construction Engineering and Management, American Society of Civil Engineering, June 1987, pp. 302-314.