Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

A CONCEPTUAL ACTIVITY CYCLE-BASED SIMULATION MODELING METHOD

Jingsheng Shi

Department of Building and Construction
City University of Hong Kong
Kowloon, Hong Kong

ABSTRACT

This paper studies activity cycle diagrams (ACD) for modeling construction processes using activities as the basic elements. An ACD is obtained by combining the activity cycles of all involving resources. Two basic symbols are introduced to graphically represent the activity cycle of a resource with incorporating resource requirements into individual activities. Then the activity cycle based modeling (ACBM) method is studied to translate the ACD statements of a process into an equivalent executable SLAM II simulation model. The translation can be automated by implementing the generalized rules into a computer system. This approach simplifies the modeling task for a construction process into defining an activity cycle diagram, which is much easier than directly constructing a simulation model. A concreting process example is used to illustrate the method.

1 INTRODUCTION

Computer simulation can model and simulate a construction process on the operational level by considering the random nature, resource-driven characteristics and dynamic interactions during operations (Shi and AbouRizk 1997). It been proven to be an effective tool for planning and improving the performance of the real world construction operations (Halpin 1977, Paulson et al 1987, and Vanegas et al 1993). Since the development of CYCLONE by Halpin (1977, 1990), construction simulation has been advancing very fast with computer technology. Various construction simulation languages have been developed such as INSIGHT (Paulson 1978), RESQUE (Chang 1987), UM-CYCLONE (Ioannou 1989), CIPROS (Tommelein and Odeh 1994), STROBOSCOPE (Martinez and Ioannou 1994). Researchers have also attempted to enhance modeling and simulation capabilities required by construction simulation. DISCO (Huang et. al. 1994) and COOPS (Liu and Ioannou 1992) provide graphical interfaces to allow the user to construct a simulation model by manipulating graphical symbols on a computer screen. Model reusability was studied by Halpin et. al. (1990) and McCahill & Bernold (1993) by developing a library consisting of standard simulation models that encompass a number of widely used construction processes. HSM (Sawhney and AbouRizk 1995) introduced a hierarchical approach for project level modeling and simulation. Shi and AbouRizk (1997) presented the resource-based modeling (RBM) methodology in order to automate the modeling process of construction simulation with capturing the properties of construction operations.

Modeling construction operations has always been hindered by the need of the user to be proficient in both simulation and construction operations (Shi and AbouRizk 1997). More research needs to be done to make simulation an easy-to-use tool for the practitioner (Ibbs 1987). Automated construction simulation integrates modeling, experimentation and decision support into one system (Shi 1995, AbouRizk and Shi 1994), and provides a feasible solution.

This paper introduces activity cycle based method (ACBM) for simplifying the modeling process of construction simulation. Activity cycle diagrams (ACD) are studied to represent construction processes with incorporating resource requirements into activities. Then a modeling engine translates the ACD into an executable simulation model.

2 ACTIVITY CYCLE DIAGRAM (ACD)

Based on operational considerations, a construction project or operation can be defined in terms of processes, and a process is a collection of activities (Halpin and Riggs 1992). An activity is a readily identifiable component of a construction process or operation. Construction progress is achieved by executing these activities.

2.1 Construction resources

Resources are the driving force in construction operations. An activity always consumes time and resources to be constructed. In the context of this paper, a resource is an integrated unit which can be allocated to activities. For example, a loader engaging in loading includes the combination of a piece of loading equipment and an operator. A resource may indicates a crew sometimes such as a welding crew consisting of a welder, equipment, and support labors. A resource always has limited available units. An ordinary haul road, for instance, should not be treated as resource unless it constrains the system's operation such as a shared one-way bridge. A resource can be represented by a statement:

Resource(unit) (1)

where *resource* is the name of the resource; and *unit* is the available quantity.

If two resources involve in executing an activity, one resource usually functions as a server defined as *service resource*, and the other accepts the service defined as *served resource*. The situation for more than two resources are similar. The difference is that more than one resource can be server(s) or costumer(s). A service resource is usually stationed, and a served resource moves around in the system to request services. For instance, trucks moves in a quarry to receive services from the loader and crusher, which are stationed servers. However, a service resource can become a served resource depending on its engaged tasks. A loader, for example, provides service to trucks in loading and receives service from mechanics in maintenance.

A resource can engage in either one single activity or a series of activities. It should be allocated before the activity starts and be released after its assigned duty is complete.

2.2 Activity cycle diagram (ACD)

Activities to be constructed by the same resource(s) can be linked together according to their logical sequence to form an activity chain, which is defined as the activity cycle of the resource. An activity cycle can be either a closed or an open loop, and has at least one resource going through it. The activity cycle of a resource may have only one activity, and it should not be considered as a separate one if this activity can be included in the activity cycle of another resource.

A construction process always has at least one physical production or processing element going through it. For example, the production element is "soil" in an earthmoving process and "concrete" in a concreting process. Multiple activity cycles can be combined into one system by following the processing sequence of production elements or information flowing direction in the process. The system constitutes an activity cycle-based representation of the process, and its graphical representation is called as activity cycle diagram (ACD).

To facilitate graphical representation, two symbols are defined as shown in Figure 1, in which *square* is used to represent an activity, and *arrow* is used to represent logical sequence. This convention is in consistence with traditional precedence network diagrams. In addition to duration as in CPM, however, required resources are also attached to activities in ACD.

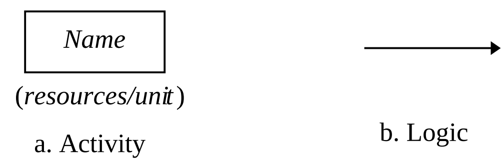

Figure 1: Basic Elements in ACD

An activity can also be equivalently represented by a statement:

Name(duration; resource/unit; followers) (2)

where *Name* is the description of the activity such as loading; *duration* is the time required to execute the activity; *resource* and *unit* are the required resource and quantity; and *followers* are the following activities.

A construction activity can be classified into either regular or service category. A regular activity does not require additional resource except what has been allocated, and can start whenever an entity arrives at it; but a service activity requires additional service resource(s) to be allocated before it can start. A service activity requires at least two resources which play the two different roles of server and costumer. However, an activity which requires two or more resources does not have to be a service activity; instead, it is still a regular activity if all required resources have been allocated in its preceding activities. For instance, considering *place_concrete* and *vibrate* two activities. *Place_concrete* is a service activity because it cannot start until concrete, a crew and a crane are all available; but *vibrate* is a regular activity because no additional resource is required except concrete and crew which are already allocated in *place_concrete*.

One round operation of a service activity may require the same involvement from all involving resources. This is defined as the even match among resources. For instance, costumer and teller in a bank belong to the even match because both the costumer and the teller are released after the service is over. On the other hand, the match may not be even when one round operation of some resource requires multiple rounds of operations of the other resource(s). This is defined as uneven match.

For example, if one loading pass is defined as loading activity, multiple passes are needed to fill one truck. To model uneven match, a fraction is used to indicate the required unit for a resource. For example, the required unit of truck in loading can be specified by 1/5 if five loading passes are required to fill one truck.

2.3 Represent a concreting process in ACD

A concreting process is adopted from Halpin (1986) to illustrate how to construct an ACD. In the process, the concrete is produced at the off-site batch plant, and transported to site by transit mix trucks. The fresh concrete arriving at the job site is moved to placement location by a hoist, held temporarily in a storage hopper and then distributed using rubber tired buggies. The resources involved in this process are plant, trucks, hoist, hopper, buggy, and crew. In the six resources, plant's operation is represented by one activity and is included in the trucks' operation cycle; and the crew's operation is included in the buggy's operation cycle. Therefore, four basic activity cycles can be defined, and can be integrated into the ACD as shown in Figure 2 by following the processing sequence of the production element (concrete).

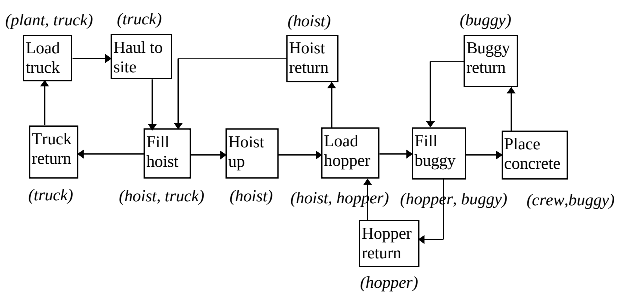

Figure 2: ACD of a Simple Concreting Process

In Figure 2, the text in each box indicates the name of the activity, and text in the nearby parenthesis indicates the required resource(s) for constructing the activity. The equivalent statements are shown in Figure 3, which starts with resources and then details all activities. Activity 3 (fill_hoist) requires 1/5 unit of hoist and one unit of truck, which means that one truck of concrete can fill five hoist buckets. The resource requirement of 1/8 unit of hopper and one unit of buggy in the statement of activity 8 (fill_buggy) indicates that the concrete of one hopper can fill 8 buggy loads.

3 ACD BASED MODELING METHODOLOGY

In the context of construction simulation, modeling is the process of representing a construction process in the syntax recognizable by the selected simulation language. A simulation model requires a more detailed description of state and movement of simulation entities than an ACD does. This section discusses how to translate the ACD representation of a construction process into an equivalent SLAM II simulation model.

```
‘ Resource
plant(1)
trucks(3)
hoist(2)
hopper(3)
buggy(5)
crew(1)

‘ Hauling cycle
1 Load_truck(uniform[0.2,0.8]; truck/1, plant/1;
haul_to_site)
2 Haul_to_site(uniform[7,10];truck/1; fill_hoist)
3 Fill_hoist(1; hoist/[1/5], truck/1; hoist_up,
truck_return)
4 Truck_return(6; truck/1; load_truck)

‘ Hoist cycle
5 Hoist_up(1; hoist/1; hoist_return, load_hopper)
6 Load_hopper(1.8; hoist/1, hopper/1; hoist_return,
fill_buggy)
7 Hoist_return(0.6, hoist/1; fill_hoist)

‘ Hopper cycle
8 Fill_buggy(0.3; hopper/[1/8], buggy/1;
hooper_return)
9 Hopper_return(1; hopper/1; load_hopper)

‘ Buggy cycle
10 Place_concrete(5.5; buggy/1, crew/1;
buggy_return)
11 Buggy_return(1;buggy/1; fill_buggy)

End
```

Figure 3: Statements of Concreting Process

3.1 Introduction to SLAMSYSTEM

SLAM II is a powerful general purpose simulation language which has a wide range of application. It provides 35 standard modeling elements, each of which models specific physical things in the real world such as activities, resources, and decision points, to facilitate the user in formulating a simulation model for a system. The basic modeling symbols used in this paper are listed in Fig. 4.

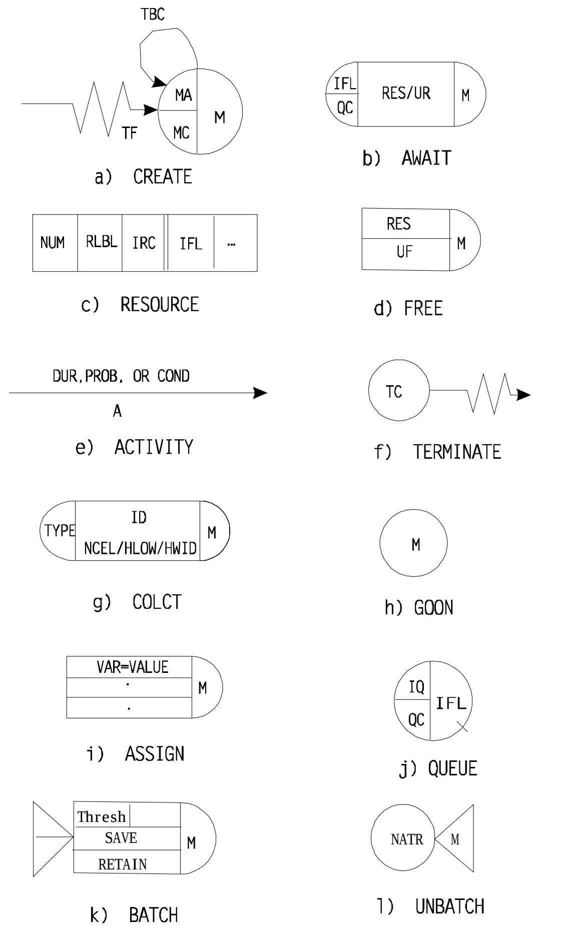

Figure 4: Part SLAM II Modeling Elements (Pritsker 1994)

The modeling functions of these elements are briefly explained as follows. A detailed description can be found in Pritsker (1994).

- CREATE: Generates entities.
- AWAIT: A queuing-type node which holds entities until its required resources are available.
- RESOURCE: It defines a resource pool and works with *AWAIT* node(s).
- FREE: Relinquish a resource which has been allocated by an *AWAIT* node.
- ACTIVITY: Models activities. An *ACTIVITY* node can also function as a link to assemble two nodes.
- TERMINATE: Destroy entities from the system.
- COLCT: Collect statistics on observations taken during simulation.
- GOON: Provide linkage for activities in series.
- ASSIGN: Prescribe values to variables or to the attributes of an entity.
- QUEUE: Model entities wait.
- BATCH: accumulated multiple entities into one.
- UNBATCH: split one entity into multiple entities.

3.2 ACD based modeling methodology

Resource and activity are the two basic elements in an ACD. Because an activity may interact with resources and other activities, it is very important to consider these relationships into the translation of ACD elements.

a. Resource

Three SLAM II elements are necessary to translate the specification and allocation of a resource. A *RESOURCE BLOCK* (Fig 4c) is used to declare the resource pool with specified capacity and requested location(s); an *AWAIT* node is used to allocate a resource to an activity; and a *FREE* node is used to release an occupied resource after the activity is complete. If a resource is engaged in a series of activities, it should be allocated before the first activity in the series starts, and be released after the last activity is complete as illustrated in Figure 5.

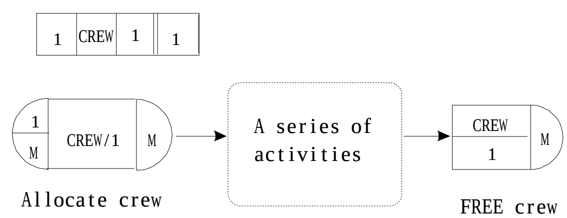

Figure 5: A Resource Specification, Allocation, and Release Example

b. Regular activity

A regular activity is graphically represented by an arrow in SLAM II (Fig. 4e). It has three functions: a) to model an activity which does not consume any resource; b) to model an activity which does not require extra resource in addition to what have occupied at its preceding activities; and c) to link two SLAM II nodes. An activity element in an ACD should be translated into a regular activity if no new resource is required in comparing with its preceding activities. In the concreting example, activity 2 (haul_to_site) is translated into a regular activity because its required *truck* has been allocated at its preceding activity 1 (load_truck).

c. Service activity under even match

If no fraction unit of a resource is specified in a service activity, all resources will be allocated with required units and be engaged in the execution of the activity. Resource is allocated by *AWAIT* node and released by *FREE* node as described previously. In the concreting process, activity 1 (*load_truck*) is a service activity under even match. One unit of the batch plant and one truck will be allocated before *load_truck* starts, graphically as modeled in Figure 6:

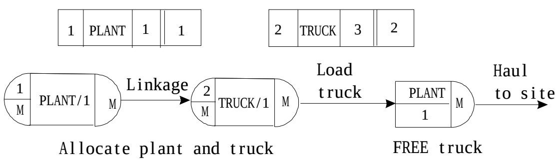

Figure 6: An Example Service Activity Under Even Match

d. Service activity under uneven match

If a fraction unit of resource is specified for a service activity, a mechanism should be constructed to balance the uneven match. An *UNBATCH* node can be used to split one entity into multiple entities, each of which matches the other resource(s); then the split entities will be grouped into one by using *BATCH* node. In the concreting example, one truck of concrete can fill five hoist buckets. The balancing mechanism can be constructed as in Figure 7.

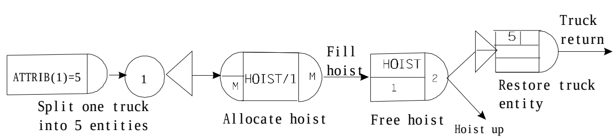

Figure 7: Uneven Matching Mechanism

This mechanism splits one truck entity into five entities, each of which matches the hoist entity separately. After the five entities have been processed by activity *fill_hoist*, a truck entity is recovered by combining the five entities into one.

e. Initiate and terminate entities

Simulation experimentation is a dynamic process in which entities flow through the model and execute functions defined by elements in the model. A simulation model is composed of inter-related activity cycles, each of which needs entities to activate its operation. A production element should be created once at its initial location, mostly at the starting activity of the starting activity cycle. If an activity cycle processes the same production element as its preceding cycle(s), no new entities should be created, and its operation can be activated by the entities released from its preceding cycle(s).

After running through the cycle, entities could be destroyed, flow to its following cycle(s), or/and go back to the starting activity of the cycle depending on the flowing direction of the production element in the cycle. If the life cycle of the production element is complete, the entities should be destroyed; it should be released to its following cycle(s) if further processing is required; and it should be routed back if a duplicate operation cycle is required. *CREATE* node can be used to initiate entities, and *TERMINATE* node can be used to destroy entities in SLAM II.

In the concreting example, the production element is “concrete”, which can be initialized before activity *load_truck* by a *create* node. The generated entity “concrete” will request the *plant* and a *truck* before *load_truck* starts as illustrate in Figure 8.

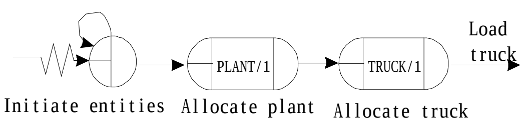

Figure 8: Initiate Simulation Entities

f. Collect simulation results

Simulation results can be collected at required locations by using *COLCT* node together with an *ASSIGN* node, in which the *ASSIGN* node calculates the value of a variable, and the *COLCT* node collects the observations of the variable.

4 MODELING EXAMPLE OF THE CONCRETING PROCESS

Applying above modeling rules, the ACD based concreting process can be translated into a SLAM II simulation model with the network diagram as shown in Figure 9, in which all resources and activities have identical labels as in the ACD statements. This model can be directly experimented in SLAMSYSTEM.

5 DISCUSSION AND FURTHER STUDIES

Activity cycle diagram (ACD) is a straightforward and effective approach of representing a construction process. It does not require more technical knowledge than CPM network does. With the developed rules which can translate an ACD into a simulation model, the construction modeling process is simplified without requiring the user to be proficient in simulation. Therefore, this study introduces a feasible solution for significantly simplifying construction simulation.

The study is currently on the conceptual level. Detailed development and implementation work is necessary to provide an easy-to-use ACD based modeling system although certain preliminary implementations have been carried out to validate the concept presented in the paper.

If a construction process gets too large or too complex, such as project level modeling, it might be very difficult or impossible to translate the ACD statements of the process into an equivalent simulation model by using the generalized rules discussed in the paper. Further studies will be conducted to integrate the activity cycle based modeling (ACBM) method with resource-based modeling (RBM) method presented by Shi and AbouRizk (1997). ACBM can be developed to generate or update atomic models required by RBM; and the coupling capabilities introduced in RBM will be substantiated to combine the involved atomic models into a working simulation model for the construction process/project. This integration will bring in an easy-to-use, fully automated, and flexible modeling system for construction simulation.

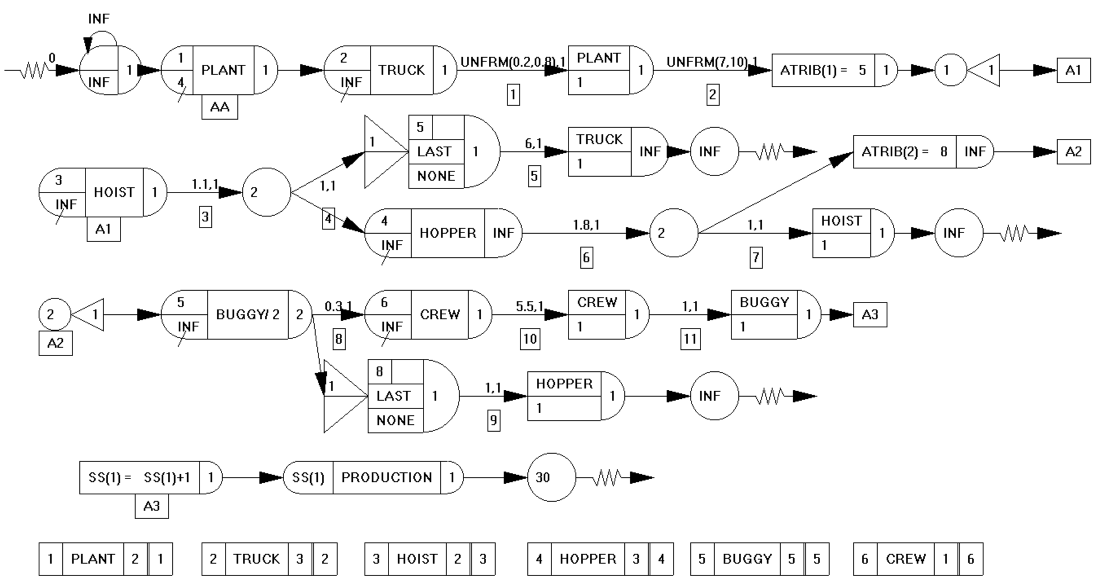

Figure 9: SLAM II Model of the Concreting Process

REFERENCES

AbouRizk, S. and Shi, J (1994). "Automated Construction Simulation Optimization.", *J. Constr. Engrg. & Mgmt*, ASCE, 120(2), 374-385.

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

Fishwick, P. (1995). "*Simulation Model Design and Execution: Building Digital Worlds: Building Digital Worlds*." Prentice Hall Inc.

Halpin, D.W. (1977). "CYCLONE - A Method for Modeling Job Site Processes." *J. Constr. Div*., ASCE, 103(3), 489-499.

Halpin, D. W. (1990). "*MicroCYCLONE User's Manual*." Division of Construction Engineering and Management, Purdue University, West Lafayette, Indiana.

Halpin, D.W. and Riggs, L. S. (1992). "Planning and Analysis of Construction Operations." John Wiley & Sons, Inc., New York, N.W.

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

Ibbs, C.W. (1987). "Future direction for computerized construction research." *J. Constr. Engrg. & Mgmt*, ASCE. 112(3), 326-345.

Ioannou, P.G (1989). "UM_CYCLONE user's guide." Dept. of Civil Engineering, The University of Michigan, Ann Arbor, Mich.

Liu, L. Y. and P. G. Ioannou (1992). "Graphical Object-Oriented Discrete-Event Simulation System." *Proceedings of the 1992 Winder Simulation Conference*, 1285-1291.

McCahill, D.F. and L.E. Bernold (1993). "Resource-Oriented Modeling and Simulation in Construction." *J. Constr. Engrg. & Mgmt*, ASCE, 119(3), 590-606.

Paulson, B.C. Jr., (1978). "Interactive Graphics for Simulating Construction Operations." *J. Constr. Div.*, ASCE, 104(1), 69-76.

Paulson, B.C., Jr., Chan, W.T., Koo, C.C. (1987). "Construction Operations Simulation by Microcomputer." *J. Constr. Engrg. & Mgmt*., ASCE, 113(2), 302-314.

Pritsker, A. A. B. (1986), "*Introduction to Simulation and SLAM-II*", John and Sons, Inc., New York, N.Y.

Sawhney, A. And S.M. AbouRizk (1995). "Simulation based planning method for construction project." *J. Constr. Engrg. & Mgmt*, ASCE, 121(3), 297-303.

Shi, J. and S. AbouRizk (1994). "A Resource-based Simulation Approach with Application in Earthmoving/Strip Mining." *Proceedings of the 1994 Winter Simulation Conference*, December 11-14, Florida, 1124-1129.

Shi, J. (1995). "*Automated modeling and optimization for construction simulation*." PhD thesis, Department of Civil Engineering, University of Alberta.

Shi, J. And S. AbouRizk (1997a). "Resource-based Modeling for Construction Simulation." Accepted for publication, *J. Constr. Engrg. & Mgmt*, ASCE.

Tommelein, I. D., and A. M. Odeh (1994). "Knowledge-based Assembly of Simulation Networks Using Construction Designs, Plans, and Methods." *Proceedings of the 1994 Winter Simulation Conference*, 1145-1158.

Vanegas, J.A., E.B. Bravo, and D.W. Halpin (1993). "Simulation Technologies for Planning Heavy Construction Processes." *J. Constr. Engrg. & Mgmt.*, ASCE, Vol.119(2), 336-354.

Zeigler, B.P. (1987). "Hierarchical, modular discrete-event modelling in an object-oriented environment." *Simulation*, 49(5), 219-230.

JONATHAN JINGSHENG SHI is an Assistant Professor in the Department of Building and Construction of City University of Hong Kong. Dr. Shi received his Bachelor and Master of Science in Civil Engineering from the Wuhan University of Hydraulic and Electric Engineering of China in 1982 and 1985 respectively. He received his Ph.D. degree from the University of Alberta in 1995. His research interests mainly include automated modeling, simulation-based decision support, automated simulation system, and other computer applications for construction. He has been actively involved in various research and consulting projects of employing simulation for analyzing and improving the performance of construction operations.