Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

PRACTICAL APPROACHES FOR VALIDATING A CONSTRUCTION SIMULATION

Jonathan Jingsheng Shi

Department of Civil and Architectural Engineering
Illinois Institute of Technology
3201 South Dearborn Street
Chicago, IL 60616-3793, U.S.A.

ABSTRACT

This paper presents three methods implemented in the activity-based construction modeling and simulation (ABC) system for validating a construction simulation. The first method reports a simulation experiment in the chronological order so that the user can examine the operating sequence of the model. The second method provides the summary information of total operating counts and mean durations of all activities in the model so that the user can evaluate whether all activities have been correctly executed during simulation. The third method produces the cyclic report of a selected resource entity so that the user can examine whether the entity is moving in the correct logical and chronological order during simulation. The three methods can effectively assist the user in debugging a simulation model so as to assure that the simulation is correctly conducted. A road paving process is employed to illustrate these methods.

1 INTRODUCTION

Computer simulation has been growing rapidly with the advancement of computer technology and has become one of the most popular techniques for practitioners in operations research and in the manufacturing industry (Pidd 1984, Paul 1991). The first application of simulation in construction was reported in 1963 (Teicholz 1963). Since the development of CYCLONE (Halpin 1977), extensive research efforts have resulted in many construction simulation systems such as INSIGHT (Paulson 1978), RESQUE (Chang 1987), UM-CYCLONE (Ioannou 1989), COOPS (Liu and Ioannou 1992), DISCO (Huang et al. 1994), CIPROS (Tommelein and Odeh 1994), STROBOSCOPE (Martinez and Ioannou 1999), Symphony (Hajjar and AbouRizk 1999), and ABC (Shi 1999). Actual applications reported significant improvement in construction productivity. An international contractor (Dragados) has used CYCLONE for over 30 projects and has recorded productivity improvements ranging from 30% to 300%. For every hour of analyst time used, a saving of $2,000 is realized (Halpin and Martinez 1999).

Simulation is the technology of experimenting on a computer with a mathematical and/or logical model built for a real world system. The results from the experiment are used for analyzing the system. Decisions about the system such as performance improvement are made based on the simulation results. However, before the simulation results can be used for this purpose, it is essential to confirm that the simulation model does characterize the real system and the simulation results are a valid representation of the system's performance. As Robinson (1996) pointed out, a simulation must be confirmed with three issues: validity, credibility, and acceptability. A simulation model must be verified and the simulation results must be validated in order to provide valid and trustful results. The credibility and acceptability of a simulation are reflected in the willingness of the user to base decisions on the results obtained. In Zeigler's terms (1984), verification is a process by which we try to assure ourselves that the simulation model is properly realized; validation is a process whereby we assess the degree to which the simulation model input:output relations map onto those of the real system.

Simulation is praised for the ability to describe a real world system in any detail as expected and the ability to address the random and dynamic features in the operation of the system. However, the complexity in employing simulation greatly increases the chances of getting incorrect or error results which may not characterize the system. Generally speaking, there are three types of errors in simulation (Pidd 1998):

- Type zero errors. This type of errors occurs when the modeler asks the wrong questions. The results are that the model does totally the wrong thing, or the model does not operate in the fashion as it is intended to. This type of errors mostly occur if the modeler does not have a good understanding about the real world system and/or about the concerned issues of the system.

- Type I errors. Type I errors correspond to the classical statistical hypothesis testing while a valid model is wrongly rejected because there is a certain probability that an error may occur.
- Type II errors. These errors occur while a false model is accepted as valid because of the accuracy of the statistics.

Type zero errors are very severe and must be avoided. Type I and II errors cannot be avoided but can be limited by a given confidence level. The major tasks in verification and validation are to identify and to fix type zero errors. Two distinct approaches are available: black-box and white-box approaches (Pidd 1998).

As the name suggests, the back-box approach treats the simulation model and the real world system as two separate black boxes. The inner workings of both are unknown but a set of observations are obtainable for both. The two sets of observations are then tested for significance in differences. The basic hypothesis is that the observations of the model should be indistinguishable from those of the real system if the simulation is valid. Although the black-box approach provides a straightforward approach for validating a simulation study, its applicability is limited for the following two reasons: 1) Observations may not be available for the real system; and 2) The selection of factors to be tested may be tricky. An arbitrary testing of some factors may not validate a simulation.

When direct observations are not available for the real system, AbouRizk et al. (1991) recommended analytical methods for estimating the values of selected parameters of the system. Basically, the user would analyze the system using an analytical method in addition to a simulation study. The results from the analytical method are compared to the results obtained from simulation. Specifically, the authors attempted the method productivity delay model (MPDM, Adrian and Boyer 1976) and queuing theory for validating the simulation results of a roof truss installation process. Productivity value was selected as the parameter to be evaluated and compared. The authors argued that the productivity values obtained from the three methods should be consistent under the same given initial conditions.

The black-box approach affirms whether the simulation results are in a reasonable range so that the user can establish confidence on the obtained results. However, the approach does not eliminate the possibility of errors occurring in simulation, especially in a simulation model and its simulation process. In other words, the consistent results from a simulation may be a coincidence.

Simulation attempts to display a real system on a computer. One obvious question is whether the simulated progress is in line with the system's progress in the real world. To assure a model to operate in the manner as intended, white-box approach aims at examining the internal structures of the model and the system so that errors are detected and are fixed. Consequently, the simulation is conducted in the correct manner.

Because of the complexity of simulation and a great amount of computation in advancing a simulation, the white-box approach can only be mastered by experienced users with comprehensive simulation knowledge accumulated from many simulation projects. Based on the ABC system (Shi 1999), this paper presents three easy-to-use methods for the user to examine a dynamic simulation progress so as to verify and validate the simulation.

2 A CONCRETING PROCESS

To illustrate the methods, a concreting process from Halpin and Riggs (1992) is used in this paper. Open bay trucks with 5 compartments are batched with five dry batches of concrete (one compartment at one time) at the batch tower. The loaded truck hauls the concrete to a pavement job site, and dumps the batches separately into the skipper of a mixer near the paving site. The mixer then mixes the concrete and dumps the wet concrete to a bucket. The bucket is then lifted to the pavement location by a crane. A crew is available for emptying the bucket and spreading the concrete. A schematic diagram of the process is shown in Figure 1. The initial resource capacity and activity durations are listed in Tables 1 and 2 respectively. Simulating the process with MicroCYCLONE, the production rate is 10 spreading cycles per hour for the crew (Halpin and Riggs 1992).

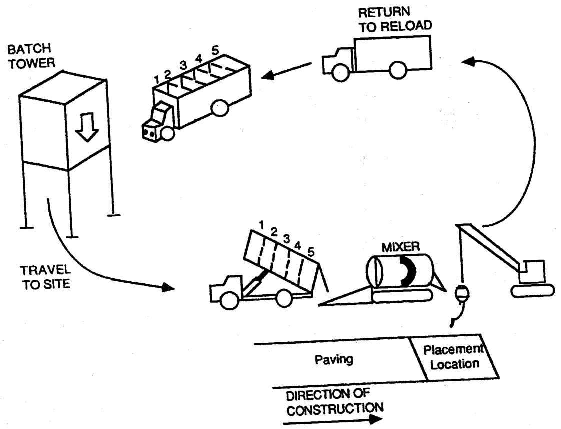


Figure 1: The Schematic Diagram of a Concreting Process (Halpin & Riggs 1992)

Table 1: Resource Capacity

Resource name (1)	Capacity (2)
Batch tower	1
Trucks	4
Mixer	1
Buckets	2
Crane	1
Crew	1

Table 2: Activity Durations

Activity name (1)	Duration in minutes (2)
Load truck	5
Travel to mixer	10
Dump to skipper	1
Truck return	8
Mix	3
Fill bucket	0.5
Swing crane	0.25
Empty bucket	0.3
Crane return	0.2
Spread conrete	5

The corresponding ABC model is depicted in Figure 2, in which the required truck unit at Activity 1 is –5 representing that a truck requires 5 loading passes to fill its 5 compartments. Running the model in the ABC system, 50 production cycles are collected at activity spread concrete in the total simulation time of 304 minutes, which is equivalent to the production rate obtained from MicroCYCLONE.

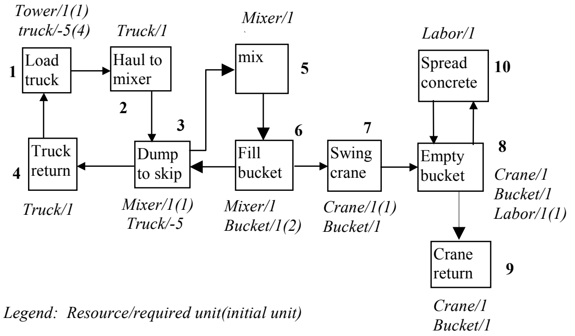


Figure 2: The ABC Model of the Concreting Process

3 WHITE-BOX BASED METHODS FOR SIMULATION VALIDATION

A simulation experiment advances the dynamic operation of the model in a simulated chronological order. The simulated progress must be in line with the actual progress of the system. To verify and validate the simulated progress, the ABC system traces the entire simulation progress of the model in the chronological order in the following format:

Tnow AL AN FT AD R/ID(IT);
where Tnow = current simulation time; AL = activity label; AN = activity name; FT = finish time; AD = activity duration; R/ID(IT) = resource name/identification number(idle time).

The ABC simulation (Shi 1999) assigns a unique identification (ID) number for a resource entity when it is first released from its initial source (either an activity or the resource pool); the entity is then identified by the ID # for its entire life cycle during simulation; and its movement in the simulation model and change of state are chronologically traced along simulation time. In other words, the ABC simulation can tell where and which state a given resource entity is at any simulation time. Multiple units of a resource are distinguished by different ID numbers. For instance, if there are three trucks in an earthmoving process, each of them is assigned with a unique ID number and follows its own path during simulation.

Each line in the tracing report details the execution information of an activity. It tells the time when the activity starts and finishes its construction, as well as the information of the involved resources. An example line is as the following:

25 1 load_truck 30 5 tower/1(0), truck/3(25);

This line tells: Simulation time is advanced to time 25; Activity 1 "load truck" starts operation; it takes 5 minutes to operate, and completes at time 30; Two resources (tower and truck) are engaged in the operation of the activity. The tower entity is labeled #1 and has zero waiting time before the activity starts. The truck entity is label #3 and has waited 25 minutes before the activity starts its operation.

The ABC system saves the simulation progress information in a file named *project_name.a#*. Here, *project_name* is the name of the model. "*a#*" is the file extension with one letter "*a*" followed by a digit representing the simulation run number (The maximal simulation run must be less than 100 so that the file extension has a maximum of three characters). The ASC II format file is saved in the same directory of the project. It provides source data for all reports. If a simulation run is terminated as expected, the file contains the progress information of the entire simulation process; otherwise, the file contains the progress information of the simulation run to the time when it is terminated.

In the following sections, we present three methods to assist the user in affirming that: 1) The operation of the model is advanced in the right chronological and logical order; 2) All activities have executed as intended; and 3) Resource entities have moved in the correct cycles during simulation.

3.1 Tracing Simulation Progress

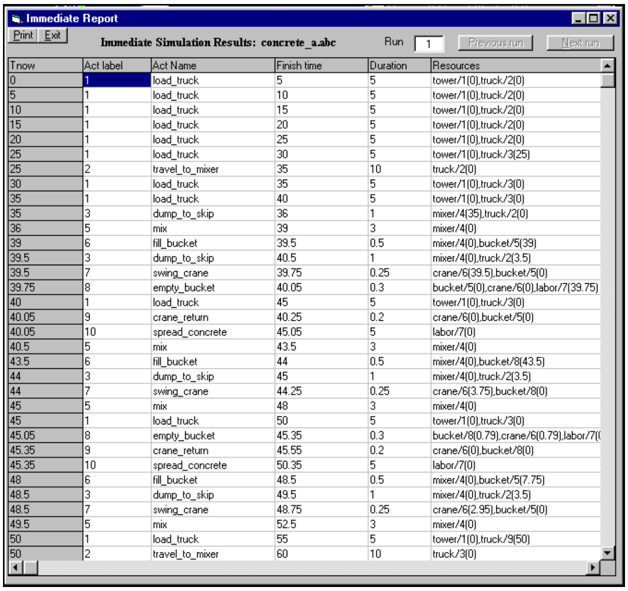
If the user knows the actual progress of a construction process under study, she/he should examine whether the simulated progress is in line with the actual progress. Specifically, activities must start in the right sequence and at appropriate times. If a model has errors, its simulation run would abnormally terminate or may not execute in the correct order. It is common to see users, especially beginners, get frustrated and do not know what to do while a simulation run gets to nowhere after so much effort in building up a model. Sometimes, a simulation cannot start or terminates after a couple of activities are executed.

An effective approach to examine a simulation experiment is to trace the simulation history. The historical information presents a tracing report in which activities are listed in the chronological sequence as they were executed. Examining the tracing report assists the user in assuring whether the simulation run is correctly conducted. For instance, if a tracing report does not have any executed activity (i.e. the simulation did not start), the user should examine the model focusing on the activities which should start at the beginning of the simulation. An activity cannot start execution for one of or two reasons: a) insufficient resources, b) logical constraints from its preceding activities. If a simulation is not completed as expected but its tracing report contains executed activities, the user should examine the model by following the sequence of activities executed and focusing on the last activities executed so as to detect why the simulation could not continue.

The ABC system provides an immediate report for any given simulation run. The tracing report of the concreting process is shown in Figure 3, which depicts that the simulation starts off with a truck (ID# 2) being loaded with 5 batches of dry concrete. While the truck is fully loaded at time 25, the next truck (ID #3) starts being loaded and the loaded truck (ID #2) starts traveling to the mixer (Activity 2). When simulation time is advanced to 30, the truck (ID #3) is still under loading with dry concrete. When time is advanced to 35, the truck (ID #2) arrives at the mixer and starts dumping the dry concrete to the skipper, and it takes one minute for dumping. The mixer starts mixing at time 36. The mixed concrete is ready after three minutes and is dumped to a bucket at time 39… Following the simulation time in the report, the user can examine the sequence of activities executed. Obviously, the progress must be consistent with the actual construction progress. Because activities are the only elements in the ABC simulation, the tracing report contains only the execution information of activities.

To examine the entire tracing report tends to be a tedious task because a simulation run may contain thousands of lines. The author finds it very helpful to examine at least the first cycle of operation of the model. The examination affirms whether activities are getting started and are advanced in the correct sequence, and whether resource entities are initialized at the right locations. Any inconsistencies indicate errors in the model. When a simulation run terminates abnormally, the tracing report pinpoints where and when the simulation terminates so that the user can investigate causes of the occurrence and fix the problems efficiently.

It is sometimes complicated even to examine the first operating cycle of a large simulation model because a large model has many concurrent sub-cycles and each sub-cycle has its own set of activities and logical sequence. The animation technology provides an effective means for the user to visualize the simulation progress on a computer screen so that the simulation progress can be visually validated (Zhang and Shi 2000).



Immediate Simulation Results: concrete_a.abc   Run 1

Tnow	Act label	Act Name	Finish time	Duration	Resources
0	1	load_truck	5	5	tower/1(0),truck/2(0)
5	1	load_truck	10	5	tower/1(0),truck/2(0)
10	1	load_truck	15	5	tower/1(0),truck/2(0)
15	1	load_truck	20	5	tower/1(0),truck/2(0)
20	1	load_truck	25	5	tower/1(0),truck/2(0)
25	1	load_truck	30	5	tower/1(0),truck/3(25)
25	2	travel_to_mixer	35	10	truck/2(0)
30	1	load_truck	35	5	tower/1(0),truck/3(0)
35	1	load_truck	40	5	tower/1(0),truck/3(0)
35	3	dump_to_skip	36	1	mixer/4(35),truck/2(0)
36	5	mix	39	3	mixer/4(0)
39	6	fill_bucket	39.5	0.5	mixer/4(0),bucket/5(39)
39.5	3	dump_to_skip	40.5	1	mixer/4(0),truck/2(3.5)
39.5	7	swing_crane	39.75	0.25	crane/6(39.5),bucket/5(0)
39.75	8	empty_bucket	40.05	0.3	bucket/5(0),crane/6(0),labor/7(39.75)
40	1	load_truck	45	5	tower/1(0),truck/3(0)
40.05	9	crane_return	40.25	0.2	crane/6(0),bucket/5(0)
40.05	10	spread_concrete	45.05	5	labor/7(0)
40.5	5	mix	43.5	3	mixer/4(0)
43.5	6	fill_bucket	44	0.5	mixer/4(0),bucket/8(43.5)
44	3	dump_to_skip	45	1	mixer/4(0),truck/2(3.5)
44	7	swing_crane	44.25	0.25	crane/6(3.75),bucket/8(0)
45	5	mix	48	3	mixer/4(0)
45	1	load_truck	50	5	tower/1(0),truck/3(0)
45.05	8	empty_bucket	45.35	0.3	bucket/8(0.79),crane/6(0.79),labor/7(
45.35	9	crane_return	45.55	0.2	crane/6(0),bucket/8(0)
45.35	10	spread_concrete	50.35	5	labor/7(0)
48	6	fill_bucket	48.5	0.5	mixer/4(0),bucket/5(7.75)
48.5	3	dump_to_skip	49.5	1	mixer/4(0),truck/2(3.5)
48.5	7	swing_crane	48.75	0.25	crane/6(2.95),bucket/5(0)
49.5	5	mix	52.5	3	mixer/4(0)
50	1	load_truck	55	5	tower/1(0),truck/9(50)
50	2	travel_to_mixer	60	10	truck/3(0)

Figure 3: Simulation Progress Report

3.2 Examining Operations of Activities

The operation of an activity is normally restrained by conditions such as preceding logics and required resources. An activity must register as many operations as expected during simulation; otherwise the model has errors associated with the activity. If logical relationships are inappropriately defined or resources are incorrectly initialized or are released to wrong locations, the simulation operation would go wrong. Let's see a simple model shown in Figure 4, in which one unit of resource "Mach" is initialized at activity AA; and one unit of resource "Crew" is initialized at activity CC. If "Mach" is wrongly released from activity AA to activity BB, the simulation will not be able to continue after activity AA is completed because activity BB needs "Crew" which is held at activity CC and activity CC is waiting for "Mach" to get started. Finally, no activity can start and the simulation terminates. To correct this problem, the user must change the release destination of resource "Mach" at activity AA from "succeeding activity" to "resource pool".

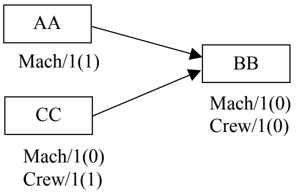


Figure 4: An Example Problem Model

The registered operating counts of activities provide an indicator to evaluate whether a simulation model has been executed as expected. Any errors in a simulation model can result in incorrect operating counts for some activities. For instance, if 30 truck loads of concrete are intended to be placed in a concreting process, relevant activities such as truck hauling and concreting placing should register 30 or corresponding operations in the simulation.

Mean activity durations also provides an indication for the user to evaluate a simulation. Incorrect entries of activity durations would significantly affect a simulation. Moreover, if a stochastic distribution is specified for an activity's duration, the mean activity duration would give the user the idea of the significance of the specification. For instance, if the loading time in the concreting example is specified with a uniform distribution in the range of 2 and 7 minutes instead of a constant 5 minutes, the simulation report shows a mean duration of 4.36 minutes collected from 50 loading operations. Is this expected?

The operating information of activities is summarized from the tracing file. The user can obtain the information from the summary report in the ABC system. The operating counts of all activities in the ABC model for the concreting process are listed in column 3 of Table 3, which shows that activities 2 and 4 have operated 10 times and all other activities have operated 50 times. Because each truck requires 5 loads to fill its box, and each truck load can feed the mixer five times, the obtained operating counts show a consistency. The obtained mean activity durations equal to the specified deterministic durations, and are listed in column 4 of Table 3.

Table 3: Operating Counts of Activities

Act. (1)	Activity name (2)	The original model		Counts of the modified model
		Counts	Dur.	
1	Load truck	50	5	50
2	Travel to mixer	10	10	10
3	Dump to skip	50	1	50
4	Truck return	10	8	10
5	Mix	50	3	50
6	Fill bucket	50	0.5	50
7	Swing crane	50	0.25	84
8	Empty bucket	50	0.3	84
9	Crane/bucket return	50	0.2	84
10	Spread concrete	50	5	84

Any inconsistency in operating counts among activities indicates errors in the simulation model. For example, if the ABC model of the concreting process is misconnected as shown in Figure 5, in which one arrow links activities 9 to 6 and another arrow links activities 9 to 7. Subjectively, the user may expect bucket entities to be returned to activity 6, and the crane entity to be routed to activity 7. With the modified relationships, we run the model again. The activity operating counts are listed in column 5 of Table 3, which shows that activities 7 to 10 have operated 84 times. Obviously, these activities should operate only 50 times. A close investigation would indicate the problem. Because the two added connections, the resource entity "bucket" short-circuits to activity 7 instead of activity 6 as expected. If the two connections are deleted, both "crane" and "bucket" entities are released to the resource pool, from which entity "bucket" is assigned to activity 6 and "crane" is released to activity 7 respectively.

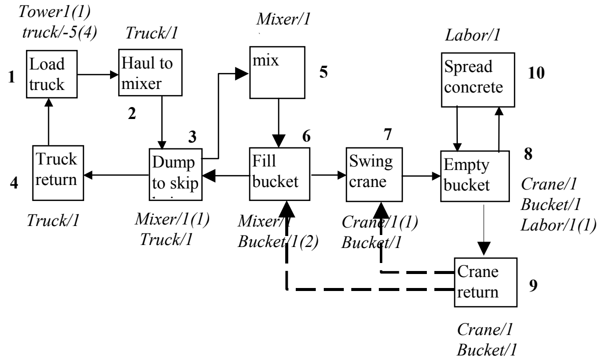


Figure 5: The Modified ABC Model of the Concreting

3.3 Examining Operating Cycles of Resource Entities

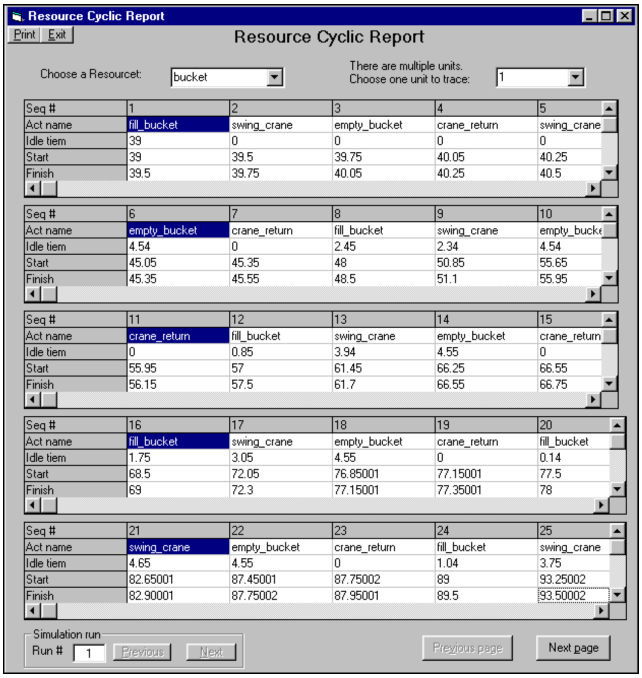
A mobile resource entity moves from one activity to another in the real world system as time advances. Its moving path usually constitutes cycles following a chronological order, typically in construction operations. For instance, the operating cycles of truck entities consist of four activities: load, haul, dump, and return in an earthmoving operation; in each cycle, the finish time of "load" is the feasible start time of activity haul; the finish time of activity haul is the feasible start time of activity dump; the finish time of return is the feasible start time of activity load of the next cycle. To assure a simulation being correctly conducted, the simulated operating cycles of resource entities must be identical as these resources engage in the real system by following the same logical and chronological order. Therefore, the cyclic reports of resource entities provide an alternative for the user to examine a simulation progress and to discover any errors in the model.

To obtain cyclic reports of resource entities, the life cycle of each resource entity must be traced in simulation. The life cycle of a resource entity starts when it is released from its initialized source (an activity or the resource pool in ABC) and terminates when the entity exits the process.

Different entities of the same resource and even the same entity may have different operating cycles during simulation. For example, if there are two loading locations labeled by "load 1" and "load 2" respectively in an earthmoving process, the operating cycles of a truck entity may include either “load 1” or “load 2” depending on the status of the model at the two locations.

The ABC system provides the cyclic reporting function for resources by allowing the user to select a resource to report. If a resource has one single unit, the cyclic report is immediately available after the resource is selected; otherwise, the user must select a unit to be reported for the resource with multiple units. The cyclic report of a resource entity lists of all activities that the entity has engaged in a chronological order during simulation. The report is generated from the tracing file. For each activity, the information includes idle time of the entity before starting the activity, start time, and finish time of the activity.

The cyclic report of bucket #1 for the modified model of the concreting process is shown in Figure 6. The entity starts off at activity 6: fill bucket – Seq. #1, succeeded by activity 7: swing crane – Seq. #2, then activity 8: empty bucket – Seq. #3, and next activity 9: crane return – Seq. #4. After that, the entity is routed to activity 7: swing crane – Seq. #5 instead of activity 6: fill bucket as it should be. The cyclic report clearly indicates where the short-circuiting occurs so that the user should remove the two links added in the modified model as shown in Figure 5.



Resource Cyclic Report

Choose a Resourcet: bucket

There are multiple units.
Choose one unit to trace: 1

Seq #	1	2	3	4	5
Act name	fill_bucket	swing_crane	empty_bucket	crane_return	swing_crane
Idle tiem	39	0	0	0	0
Start	39	39.5	39.75	40.05	40.25
Finish	39.5	39.75	40.05	40.25	40.5

Seq #	6	7	8	9	10
Act name	empty_bucket	crane_return	fill_bucket	swing_crane	empty_bucke
Idle tiem	4.54	0	2.45	2.34	4.54
Start	45.05	45.35	48	50.85	55.65
Finish	45.35	45.55	48.5	51.1	55.95

Seq #	11	12	13	14	15
Act name	crane_return	fill_bucket	swing_crane	empty_bucket	crane_return
Idle tiem	0	0.85	3.94	4.55	0
Start	55.95	57	61.45	66.25	66.55
Finish	56.15	57.5	61.7	66.55	66.75

Seq #	16	17	18	19	20
Act name	fill_bucket	swing_crane	empty_bucket	crane_return	fill_bucket
Idle tiem	1.75	3.05	4.55	0	0.14
Start	68.5	72.05	76.85001	77.15001	77.5
Finish	69	72.3	77.15001	77.35001	78

Seq #	21	22	23	24	25
Act name	swing_crane	empty_bucket	crane_return	fill_bucket	swing_crane
Idle tiem	4.65	4.55	0	1.04	3.75
Start	82.65001	87.45001	87.75002	89	93.25002
Finish	82.90001	87.75002	87.95001	89.5	93.50002

Simulation run
Run # 1

Figure 6: Cyclic Report of Bucket #1

4 CONCLUSION

The simulation technology provides a quantitative approach for studying a real world system based on a mathematical/logical model. Only valid, credible, and correct simulation results are acceptable and useful for improving a system’s performance. The black-box approach is helpful for assisting the user in determining whether the obtained simulation results are in a reasonable range by comparing them to the actual observations of the actual system or the results derived from analytical methods.

On the other hand, the white-box approach provides a means for the user to investigate how a simulation experiment is conducted. In order to get the correct results, the simulated progress based on the model must be consistent to the actual progress of the real system. Specifically, this paper presented three methods to assist the user in verifying and validating a simulation. The three methods are implemented in the ABC system with corresponding reports. By examining these reports, the user can discover errors in the model and assure a simulation being correctly conducted.

REFERENCES

AbouRick, S.M., D.W. Halpin, and S.L. Hill 1991. Measuring Productivity and Validating Microcomputer Simulation. *Microcomputers in Civil Engineering*, 6(2): 202-215.

Adrian, J. & L.T. Boyer 1976. Modeling method-productivity. *J. of Construction Division*, ASCE. 103(2): 154-168.

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

Hajjar, D. S.M. AbouRizk 1999. Simphony: An Environment for Building Special Purpose Construction Simulation Tools. *Proceedings of the 1999 Winter Simulation Conference*. IEEE, Piscataway, N.J: 998-1006.

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

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

Halpin, D.W., and L.H. Martinez 1999. Real World Applications of Construction Process Simulation. Proceedings of the 1999 Winter Simulation Conference. IEEE, Piscataway, N.J. 956-962.

Huang, R., A.M Grigoriadis, and D. W. Halpin 1994. Simulation of Cable-stayed Bridges Using DISCO. *Proceedings of 1994 Winter Simulation Conference*. IEEE, Piscataway, N.J. 1130-1136.

Ioannou, P.G 1989. UM_CYCLONE User's Guide. Dept. of Civil Engineering, The University of Michigan, Ann Arbor, Mich.

Law, A.M. and W.D. Kelton 1991. *Simulation Modeling and Analysis*. 2nd edition. McGraw-Hill, Inc. New York.

Liu, L. Y. and P. G. Ioannou 1992. Graphical Object-Oriented Discrete-Event Simulation System. *Proceedings of 1992 Winter Simulation Conference*. IEEE, Piscataway, N.J. 1285-1291.

Martinez J.C. and Ioannou P.G. 1999. General Purpose Systems for Effective Construction Simulation. *J. of Constr. Engrg. and Mgmt.*, ASCE, 125(4): 265-276.

Paul R.J. 1991. Recent Development in Simulation Modelling. *J. Opl Res. Soc.* 42(3): 217-226.

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

Pidd, M. 1984. Computer simulation for operational research in 1984. *Developments in Operational Research* (R.W. Eglese and G.K. Rand, Eds). Pergamon Press, Oxford.

Pidd, M 1992. *Computer simulation in management sciences*. 3rd edition. Chichester: John Wiley and Sons.

Pidd, M. 1998. Computer simulation for operational research in 1984. *Developments in Operational Research* (R.W. Eglese and G.K. Rand, Eds). Pergamon Press, Oxford.

Pritsker, A. A. B., O'Reilly, J.J., and LaVall, D.K. 1997. *Introduction to Simulation and SLAM-II.* 2nd edition. John and Sons, Inc., New York, N.Y.

Robinson, S.W. 1996. Service quality management in the process of delivering a simulation study. *The 14th Triennial Conference of the International Federation of OR Societies*, July 8-12, Vancouver, B.C.

Shi, J. 1999. Activity-Based Construction (ABC) Modeling and Simulation Method." *Journal of Construction Engineering and Management*, ASCE. 125(5): 354-360.

Teicholz, P. 1963. A Simulation Approach to the Selection of Construction Equipment. Tech. Rep. No. 26, The Construction Institute, Stanford University, Palo Alto, Calif.

Tommelein, I. D., and A. M. Odeh 1994. Knowledge-based Assembly of Simulation Networks Using Construction Designs, Plans, and Methods. *Proceedings of Winter Simulation Conference.* IEEE, Piscataway, N.J. 1145-1158.

Zhang, H. and J. Shi 2001. Iconic Animation for Activity-based Construction Simulation. Accepted for publication with condition, *ASCE Journal of computing in civil engineering*.

Zeigler, B.P. 1984. *Multifaceted Modelling and Discrete Event Simulation.* Academic Press, London and Orlando, Fla.

AUTHOR BIOGRAPHY

JONATHAN JINGSHENG SHI is an Associate Professor in the Department of Civil and Architectural Engineering at Illinois Institute of Technology in Chicago. He received his Ph.D. in Civil Engineering from the University of Alberta in 1995. His research interests in simulation are focused on automated modeling, simulation, and optimization in construction applications. His other research interests include neural networks, construction scheduling and planning method, and construction productivity and improvement. His email and web addresses are `<jonathan.shi@iit.edu>` and `<http://www.iit.edu/~jshi>`.