Generic Process Mapping and Simulation Methodology for Integrating Site Layout and Operations Planning in Construction

Ming Lu, M.ASCE[1]; Wah-Ho Chan[2]; Jian-Ping Zhang[3]; and Ming Cao[4]

Abstract: The present research is intended to address dynamic construction-process simulation methods, with a focus on how to effectively model resource transit among various activity locations in the site system. Following a review of basic simulation paradigms and recent research developments, we propose a new process mapping and simulation methodology for modeling construction operations. The simulation algorithm is presented and the process mapping procedure is illustrated step by step using an earth-moving example featuring technology and resource constraints. It is straightforward to convert the resultant process mapping model describing workflows and resource flows over site locations into a simulation model. A STROBOSCOPE model is formed for the same problem definition to contrast and cross-validate our methodology with the established activity cycle diagram-based modeling approach. One additional case of modeling the concreting site operations by the hoist and barrow method is also given to demonstrate the application of the proposed methodology in practical settings.

DOI: 10.1061/(ASCE)0887-3801(2007)21:6(453)

CE Database subject headings: Computer aided simulation; Simulation models; Computer models; Construction methods; Construction management; Integrated systems.

Introduction

Discrete-event simulation is a powerful method to imitate the behavior of a real-world system over time by modeling repetitive processes in which durations of operations are stochastic and many resources interact (Law and Kelton 2000). Simulation keeps track of the changes of the state of a system occurring at discrete points of time (Pidd 1998) and builds a logical model of a system for experimenting on a computer (Pritsker 1986). The statistical data generated from the experiments provide modelers with insight into a system's resource applications, interactions, and constraints (Tommelein et al. 1994).

The simulation methodology of activity-cycle diagrams (ACD) lends itself well to modeling construction operations. With the modeling capabilities and ease of use being continually enhanced, ACD-based construction-simulation tools have evolved from the original cyclic-operation-network (CYCLONE) methodology (Halpin 1977) to the programmable state and resource based simulation of construction processes (STROBOSCOPE) (Martinez 1996). However, the use of simulation in construction practices has generally been random and sporadic, and numerous attempts to interest major construction companies in simulation as a productivity-enhancing means have proved unsuccessful (Halpin 1998).

The simplicity and computerization of the critical-path method (CPM) has led to its wide adoption in construction planning. Nonetheless, CPM falls short on adequately modeling the resource flows over the site layout, which is deemed indispensable to a scheduling tool in representing the short-term, dynamic characteristics of site space use (Choo and Tommelein 1999). Still, CPM underlies the state of the art in four-dimensional modeling (Koo and Fischer 2000; Chau et al. 2004), space-constrained scheduling (Thabet and Beliveau 1997), and time–space conflict analysis in construction planning (Akinci et al. 2002). On the other hand, the lack of an accepted practical simulation technique also makes it difficult to choose objectively between possibilities in evaluating the efficiency of a site layout (Li et al. 2001). Site-layout planning research largely relies on the use of mathematical programming formulations based on user-defined "closeness" relationship values (weights) and unit transportation costs (e.g., Hegazy and Elbeltagi 1999; Mawdesley et al. 2002).

Tommelein et al. (1994) pointed out: (1) existing construction-simulation implementations did not represent many of the relevant characteristics of project components or construction resources, and (2) it is tedious to collect and assemble all required input data and to construct simulation networks. The difficulty of promoting simulation in construction is also attributed to the analytic aspect of the technique itself and the time required in learning and applying simulation tools (Halpin 1998; Shi and AbouRizk 1997; Oloufa et al. 1998). Being able to harness the power of simulation requires construction practitioners to invest

[1]Assistant Professor, Dept. of Civil and Structural Engineering, Hong Kong Polytechnic Univ., Hong Kong, China. E-mail: cemlu@polyu.edu.hk

[2]MPhil Candidate, Dept. of Civil and Structural Engineering, Hong Kong Polytechnic Univ., Hong Kong, China.

[3]Professor, Dept. of Civil Engineering, Tsinghua University, Beijing, China.

[4]Research Assistant, Dept. of Civil and Structural Engineering, Hong Kong Polytechnic Univ., Hong Kong, China.

Note. Discussion open until April 1, 2008. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on February 24, 2006; approved on March 16, 2007. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 21, No. 6, November 1, 2007. ©ASCE, ISSN 0887-3801/2007/6-453–462/$25.00.

considerable time and dedicated learning in: (1) Developing cognitive skills of observing, analyzing, and depicting site operations as needed for simulation modeling; and (2) mastering special knowledge of computer use, software specifications, and statistics (Lu and Wong 2007).

In recent years, a great deal of research has been undertaken to bridge the gap between research and application in construction simulation by simplifying simulation methodologies while retaining its modeling functionalities. Representative developments include: (1) Resource-based approaches, which generate full-scale, complex simulation models through linking atomic models for particular resource operating processes (Shi and AbouRizk 1997) or preprogrammed construction resources (Oloufa et al. 1998); (2) activity-based approaches, which mimic the commonly practiced CPM in construction planning by reducing modeling constructs of general-purpose simulation tools to activity blocks (Shi 1999; Lu 2003); and (3) the special purpose simulation approaches, which develop object-oriented simulation constructs and modeling environments native to specific construction domains so as to allow a domain expert—being a construction engineer—to conduct simulation studies with minimal learning time (Hajjar and AbouRizk 2000; Mohamed and AbouRizk 2005; Song and AbouRizk 2006).

Martinez and Ioannou (1999) examined the characteristics of discrete-event simulation systems commonly used in construction and grouped them into three general strategies: Activity scanning (AS) and process interaction (PI) are identified as the two dominant strategies, while event scheduling (ES) is viewed as an accessory to the former two categories. Different simulation strategies view a real-world system from different perspectives (Pidd 1998). Thus, the underlying strategy has a strong impact on the thought process that leads to simulation-model development, as well as on the way a model is presented to the computer (Evans 1989).

The present research is intended to address the methods for dynamic construction-process simulation, with a focus on how to effectively model resource transit among various activity locations in the site system. In particular, we enhance the simplified discrete-event simulation approach (SDESA, Lu 2003) into a generic process mapping and simulation methodology by incorporating site operations and layout planning without sacrificing the simplicity of the original SDESA model. In the remainder of this paper, we first compare AS, PI, and SDESA to expose the limitations of each in terms of modeling dynamic resource transit. Next, we enhance the SDESA algorithm and augment its model definitions, followed by the presentation of a process-mapping procedure that facilitates the creation of a SDESA model. As illustrated with an earth-moving example featuring technology and resource constraints, we go through the procedure step by step to generate a process-mapping model that describes the workflows and resource flows over site locations. The resulting process mapping model can be readily transfigured into a SDESA simulation model. One practical case of concreting a slab by the hoist-and-barrow method is also given to further demonstrate the application of the proposed methodology.

Activity Scanning Simulation Paradigm

Building a simulation model by the AS paradigm entails identifying activities and the conditions under which they take place. Hence, AS paradigm lends itself well to modeling a complex operations system in which many resources with distinct properties must collaborate to trigger activities (Hooper 1986). As simulation advances, the AS executive scans activities for evaluating activity-start conditions, and then executes those that are due to happen. Standard modeling procedures include the following steps: (1) identifying the activities in the system; (2) listing the start-up conditions for each activity; (3) drawing activities in blocks and conditions in circle shapes; (4) linking activity blocks and condition circles according to the system logic; and (5) initializing the system by assigning simulation entities into the condition circles. The resulting schematic model consisting of activity blocks, condition circles, and connection arrows is the ACD. The ACD model is conducive to the understanding and communication of a complicated, interactive construction problem.

Being a typical AS simulation approach, CYCLONE (developed for construction-operations simulation by D. W. Halpin at the University of Maryland in the early 1970s) is the most widely employed simulation methodology in construction research. CYCLONE uses a small set of basic modeling elements to map resource-driven construction processes. For instance, in a CYCLONE model, a grouping of que nodes (a circle with a slash at the lower right) and combi nodes (a rectangle with a slash at the upper-left corner) are used to trace the active and idle states of construction resources that are engaged in various activities. Based on the blueprint of CYCLONE, numerous research efforts have resulted in innovative construction-simulation methodologies and computer tools, including INSIGHT (Kalk 1980), MICROCYCLONE (Lluch and Halpin 1982; Halpin 1989), UM-Cyclone (Ioannou 1988), DISCO (Huang and Halpin 1993), and STROBOSCOPE (Martinez and Ioannou 1999). The more recent "offspring" of CYCLONE is STROBOSCOPE (Martinez 1996), which is a programmable and extensible simulation system capable of modeling the granularities of complex construction operations through writing proprietary computer code. Chua and Li (2001) developed a modeling method for portraying resource interactions on top of STROBOSCOPE.

On the other hand, some inherent disadvantages of the AS modeling strategy have hindered the wide adoption of ACD-based simulation methods by practitioners for operations planning and site-layout planning in the field. For instance, all resources, which are represented as simulation entities flowing around the model, are treated as being interchangeable. This, in part, accounts for the difficulty of distinguishing resources in a CYCLONE model, as the meaning of an entity is determined by the queue node but not the entity itself. Zhang et al. (2005) described the integration of a cell-space model with a CYCLONE simulation model for bridge-redecking operations. The cell-space model divides space into cells and the change of each cell's state over time reflects the space occupancy by a resource or an activity. As pointed out by Zhang et al. (2005), a tight coupling of CYCLONE with the space model is difficult due to the absence of a site-layout representation in a CYCLONE's schematic model, and the lack of flexibility for assigning attributes to resource entities in a CYCLONE model.

For further clarification, let us consider the following situation as an example: Two production activities share one resource, "RE_1," in handling ten jobs each, and four distinct locations in the site are involved (identified as Locations 1, 2, 3, and 4). The production activity, "Prod_11," occurs at Location 1 while the activity, "Prod_33," takes place at Location 3. At the end of handling one job by Prod_11, RE_1 will be released at Location 2, while upon finishing activity Prod_33, RE_1 will stay at Location 4. If the distances between the four locations are negligible, a CYCLONE model can be readily formed as in Fig. 1, with a que node linked with two combi nodes, meaning that a

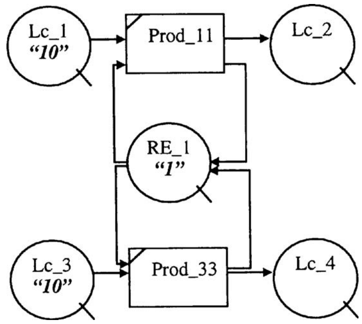

Fig. 1. CYCLONE model for one resource serving two production activities involving four locations, ignoring resource's transit between four locations

shared resource RE_1 is to serve Prod_11 and Prod_33. Each production activity is to handle ten job units as initialized in the que nodes "Lc_1" and "Lc_3." The two activities finish at Locations 3 and 4, as denoted with que nodes "Lc_2" and "Lc_4."

In many practical cases, ignoring the state changes of resource entity RE_1 in moving between the four different locations will compromise the accuracy of the model. If we consider the resource's transit processes between serving the two activities, six normal activities will have to be added to the original CYCLONE model, causing the model's size and complexity to grow considerably as shown in Fig. 2. In connection with the production activity Prod_11, activity "Trans_12" is to model the resource' s transit from Location 1 to Location 2, while "Trans_21" denotes the resource's transit from Location 2 to Location 1. Que nodes RE_1 and RE_2 represent the resource' s idle state at Locations 1 and 2, respectively. One additional node, "D_2," can be taken as a decision branch, which routes the resource to either Location 1 or 3. The lower part of the model is related to the production activity Prod_33, which can be interpreted in a similar way.

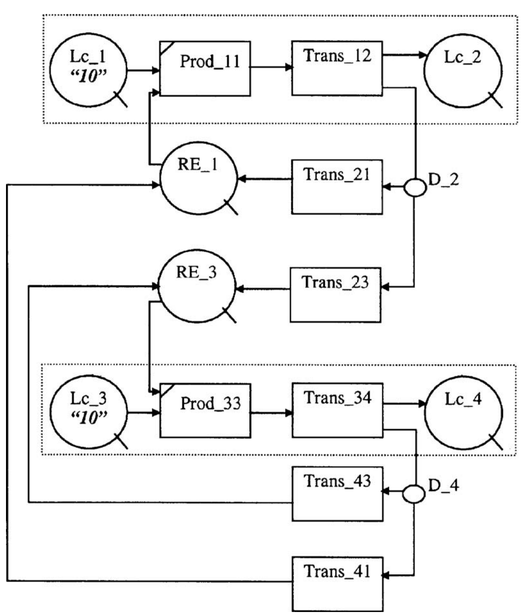

Fig. 2. CYCLONE model for one resource serving two production activities involving four locations, considering resource's transit between four locations

Process-Interaction Modeling Paradigm

Simulation methodologies based on the process-interaction (PI) modeling paradigm have made a significant impact on the manufacturing sector by bringing about tremendous cost savings and marked productivity improvement (Law and Kelton 2000). A typical manufacturing system is situated in an enclosed, indoor-factory setting. The moving entities in the form of parts or subassemblies are transported from work station to work station by material-handling systems (MHS) (such as, forklifts, conveyor systems, automated guided-vehicle systems, automated storage and retrieval systems, bridge cranes, and robots). Note that flexible, easy-to-use MHS modules are provided in most manufacturing simulation packages for rapid model development, as accurate representations of the states and patterns for those MHS can be difficult and time consuming (Law and Kelton 2000).

In contrast, any machinery or manpower resources other than MHS are deemed relatively stationary in a manufacturing system, either fixed at a certain location or moving between locations in such close proximity that the resource transit times are negligible in modeling. The common practice of manufacturing simulation modeling is to release a shared resource back to the resource pool and switch the resource's status from "occupied" to "available" immediately upon finishing the current activity. Thus, a PI-based, manufacturing-oriented simulation tool is also inadequate or inflexible to model a complex construction system, often resulting in cumbersome, intertwined network models due to the use of numerous arcs (or arrows) for directing resource flows between different site locations and activities (Lu and Wong 2007). Interested readers may refer to Lu and Wong (2007) for a detailed contrast of manufacturing versus construction systems in terms of simulation-modeling requirements.

Activity-Based Construction-Simulation Method

With the objective of making the simulation of construction operations as easy as applying critical-path scheduling, Shi (1999) developed the activity-based-construction (ABC) modeling methodology by using one modeling element of "activity" to portray the activity-cycle diagram. At the core, ABC implements the PI strategy by differentiating processing entities and resource entities and using the centralized resource pool to model any shared resources. Yet, ABC carries the same shortfall of other PI-modeling methods in coping with how to represent a resource's transit between various locations. Upon completing one activity, resource entities—which are shared by various activities—are generally released to the resource pool before being reallocated; as such, it is difficult and complicated to model the transit duration of resources between different activity locations with ABC. Zhang et al. (2002) mounted an animation layer on the ABC to visualize the queuing status and dynamic resource movements with icons, which, however, does not enhance the accuracy and flexibility of ABC with regard to modeling resource transit between site locations.

Simplified Discrete-Event Simulation Approach

To adapt PI to better cater to construction-simulation needs, Lu (2003) proposed the SDESA, which tackles the simulation of a construction system by: (1) delineating major workflows; (2) defining the activities within each workflow and the flow entities

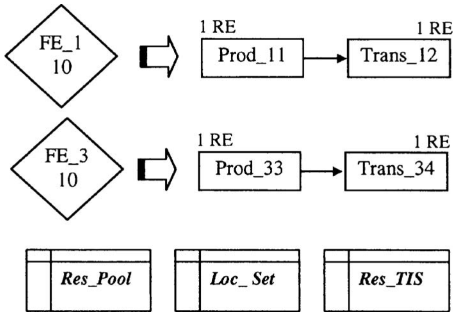

Fig. 3. Counterpart SDESA model for the above illustrative case

associated with each workflow; and (3) identifying resource entities involved in the system. The main characteristics of SDESA are summed up below.

- Differing from the manufacturing-oriented PI, workflows in SDESA are not limited to linear processes (i.e., production-line workflow), and accordingly, the associated flow entities are not limited to product units (e.g., units of material, parts, and subassemblies). Some construction resources (e.g., vehicles) can be readily identified as flow entities undergoing a close looping of activities (i.e., vehicle-loop workflow), as long as such resources are interchangeable and are bound to one workflow instead of being shared by multiple workflows. The vehicle-loop workflows are commonplace in construction and also constitute basic resource cycles in forming a CYCLONE model. Flow entities associated with all the workflows in a model are organized into one queuing structure—the flow-entity queue, which is dynamically manipulated by the simulation executive according to the SDESA algorithm.
- In a SDESA simulation model, resource entities are classified into nondisposable (manpower/machinery resources) and disposable resources, which are material or information units that are generated by one activity and requested by another; they can constitute part of resource-availability constraints in matching resources for invoking activities. All resources are organized in the resource-entity queue of the model, which is the equivalent of the resource pool in a PI model. Note SDESA uses the disposable resources to logically connect multiple workflows in a construction system.
- The SDESA executive program marshals two dynamic queuing structures (namely, the flow-entity queue and the resource-entity queue) on first-in-first-out basis, so as to advance the simulation clock and execute activities that satisfy the logical and resource-availability constraints as specified by the modeler in the network diagram model.

Fig. 3 shows the counterpart SDESA model for the above illustrative case. As for Prod_11 (on the top), ten flow entities are initialized at a diamond block, linked to Prod_11, which is followed by Trans_12. Note that Prod_11 is associated with Location 1 in the site, while Trans_12 starts at Location 1 and ends at Location 2. "1 RE" is marked on the upper-left corner of Prod_11, representing the resource requirement by Prod_11. On the other hand, "1 RE" is marked on the upper-right corner of Trans_12 to indicate the release of the resource at Location 2 upon finishing Trans_12. The lower part of the model is related to activities Prod_33 and Trans_34 from Location 3 to Location 4. The one resource entity (RE) available is initialized in the resource pool, "Res_Pool," of the SDESA model.

It is noteworthy that SDESA shares one common weakness with CYCLONE, ABC, or any PI approach: Activity locations are not part of the model definition; hence, it is still difficult to model a resource's transit between two different workflows that may take place at various locations. For instance, after processing one job unit at the end of activity Trans_12, the RE is released at Location 2. Before engaging with the next job on Prod_11 or Prod_33, the resource needs to transit from its current Location, 2, to either Locations 1 or 3. Yet, such state changes of the resource in terms of time and space cannot be readily modeled by SDESA.

The present research is intended to enhance SDESA by incorporating site operations and layout modeling while not sacrificing the simplicity of the original SDESA model. Our solution is to add two additional objects to the SDESA model definition instead of inserting extra activities and arcs. One is called "location set," (or "Loc_Set" in short) which contains the definition of main locations in the site system (such as, location's ID and its center coordinates); the other is called "resource transit information system," (or "Res_TIS" in short) which holds the transit-duration definitions for particular REs to move from one location to another. Table 1 gives the data structure of Res_TIS. In addition, the activity definition in SDESA expands to include the "start location" and "end location," while the "current location" attribute adds to the RE's existing attributes (e.g., "available time"). Note, Res_TIS contains only resource-transit information not represented within the SDESA network diagram model. In our case, Res_TIS defines the resource-transit duration: (1) from Location 2 to 1; (2) from Location 4 to 3; (3) from Location 2 to 3; and (4) from Location 4 to 1. On the other hand, the resource-transit times from Location 1 to 2 and from Location 3 to 4 have been well specified in two activity definitions in the SDESA diagram model, namely Trans_12 and Trans_34.

Accordingly, the algorithm of the SDESA executive program is enhanced and its flowchart is shown in Fig. 4, with major changes highlighted. In determining the resource available time (AVT), given that all the resources as requested by the current activity are available in the resource entity queue, if the "current location" of one RE differs from the "start location" of the current activity, the SDESA executive checks Res_TIS for the corresponding transit time (TRT) to update the available time (AVT′)

Table 1. Data Structure for Res_TIS (Resource Transit Information System)

Attribute	Description	Remarks
RE_TYPE	Type code of a RE	Linked to the RE's definition
From_Loc	Origin-location ID	Starting location of the RE linked to the site-layout breakdown structure of "Location_Set"
To_Loc	Destination-location ID	Ending location of the RE's transit, linked to "Location_Set"
TRT	Time required to transit the distance from origin location to destination location	Similar to the activity-duration definition, which can be a user-defined statistical distribution, or linked to RE's attributes

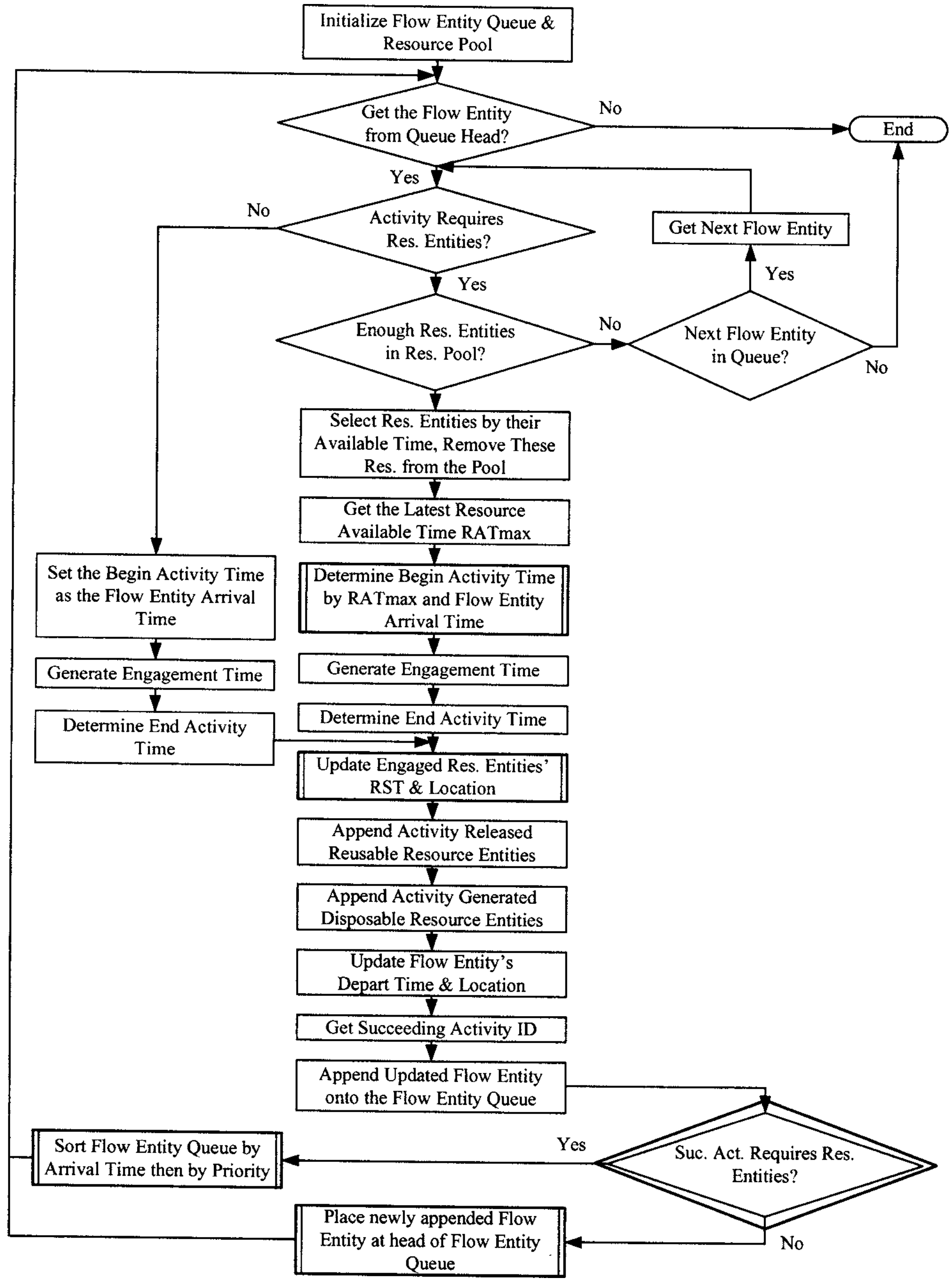

Fig. 4. Flowchart of the algorithm of the SDESA executive program

of the RE involved as Eq. (1). Additionally, its current location (RELoc) is set as the start location of the current activity (ASLoc), as Eq. (2)

$$\mathrm{AVT}' = \mathrm{AVT} + \mathrm{TRT} \tag{1}$$

$$\mathrm{RELoc}(i) = \mathrm{ASLoc}, i \in (\mathrm{resource\_entities\_involved}) \tag{2}$$

Also, considering the flow entity's arrival time (ART) at the current activity, the begin time (BT) of activity can be determined with Eq. (3)

$$\mathrm{BT} = \max\{\mathrm{ART}, \mathrm{AVT}(i)\}, i \in (\mathrm{resource\_entities\_involved}) \tag{3}$$

The activity duration (DUR) is sampled from the predefined activity duration distribution. The end time (ET) of the current activity is calculated by Eq. (4)

$$\mathrm{ET} = \mathrm{BT} + \mathrm{DUR} \tag{4}$$

At the end of the activity, the current locations of both the flow entity (FELoc) and REs released (RELoc) will be set as the current activity's finish location (AFLoc), by Eqs. (5) and (6)

$$\mathrm{FELoc} = \mathrm{AFLoc} \tag{5}$$

$$\mathrm{RELoc}(i) = \mathrm{AFLoc}, i \in (\mathrm{resource\_entites\_released}) \tag{6}$$

AVT of all the released REs and any disposable REs generated will also be updated to ET by Eqs. (7) and (8)

$$\mathrm{AVT}(i) = \mathrm{ET}, i \in (\mathrm{resource\_entities\_released}) \tag{7}$$

$$\mathrm{AVT}(j) = \mathrm{ET}, j \in (\mathrm{disposable\_resource\_entities\_generated}) \tag{8}$$

At the end of processing the current activity, the flow entity is scheduled to arrive at the succeeding activity according to the precedence relationships specified in the model definition. In comparison with the original SDESA executive algorithm (Lu 2003), the enhanced SDESA also checks on the resource requirements of the succeeding activity upon completion of processing the current activity; as shown in the lower portion of Fig. 4, the flow entity arriving at a bound activity that requires no resources (such as, Trans_13 in our case) will be appended at the head of the flow-entity queue for immediate processing. If the succeeding activity does require resources, the arriving flow entity will then be inserted into the flow-entity queue, which will be sorted first by arrival time, then by activity priority in ascending order before running the flow entity at the head of the queue through the next round of simulation processing (Lu 2003). The simulation ends when no more flow entity can be further selected and processed.

Note, in the situation of two activities competing for one available resource, by default, the SDESA executive allocates the resource to the flow entity with the earliest arrival-at-activity time; in case of identical arrival times, user-specified activity priority breaks the tie. To enable more sophisticated resource allocation control, we have also added control variables, resource attributes, and logic scripting in implementing the enhanced SDESA algorithm into a prototype computer system. Nevertheless, the most important step of applying SDESA simulation is to generate the process-mapping model, which describes workflows and resource flows over site locations. The process-mapping model can be readily transfigured and enriched into more sophisticated SDESA simulation models. The remainder of this paper presents a systematic modeling procedure that is suitable for construction practitioners to follow in representing common constraints in site-operations planning with a SDESA-compatible process-mapping model. The modeling procedure is illustrated with an earth-moving example featuring technology and resource constraints, which are characteristic of a construction system.

Modeling Procedure

To elucidate on the terminology definition and the modeling procedure of SDESA, let us consider a simple earth-moving case: At the cut, a pusher and a scraper work together to push-load one soil unit (i.e., a scraperful of soil) into the scraper's bowl. The pusher then backtracks from the "push finish point" to the "push start point" for loading the next scraper, and the scraper hauls, dumps, spreads a soil unit at the fill, and then returns to the cut. Once 20 push-loads are completed, the pusher moves to trim the side. After side trimming, the pusher then moves back from the side to the "push start point" to continue push-loading scrapers. Each scraper handles a soil unit of 20 m^3. The objective is to find the optimum number of scrapers that match one pusher tractor in moving 10,000 m^3 from the cut to the fill.

The general steps for process mapping and simulation modeling with SDESA are as follows:

1. Depict main workflows in the construction system by identifying the flow entities for each workflow and circle the associated key locations (location circles).

Two workflows can be identified for this case, namely, "scraper workflow" and "side-trimming workflow." A scraper is the flow entity that moves dirt from the cut to the fill in a cyclic workflow, while the pusher is the resource that assists the scraper in loading dirt. In addition, the pusher is also engaged in side trimming—once every 20 push-loads completed. Hence, given a total of 500 push-loads (i.e., 10,000 m^3 divided by 20 m^3 per scraperful), there are 25 side-trimming needs throughout the earth-moving operation, each being a flow entity going through the side-trimming workflow.

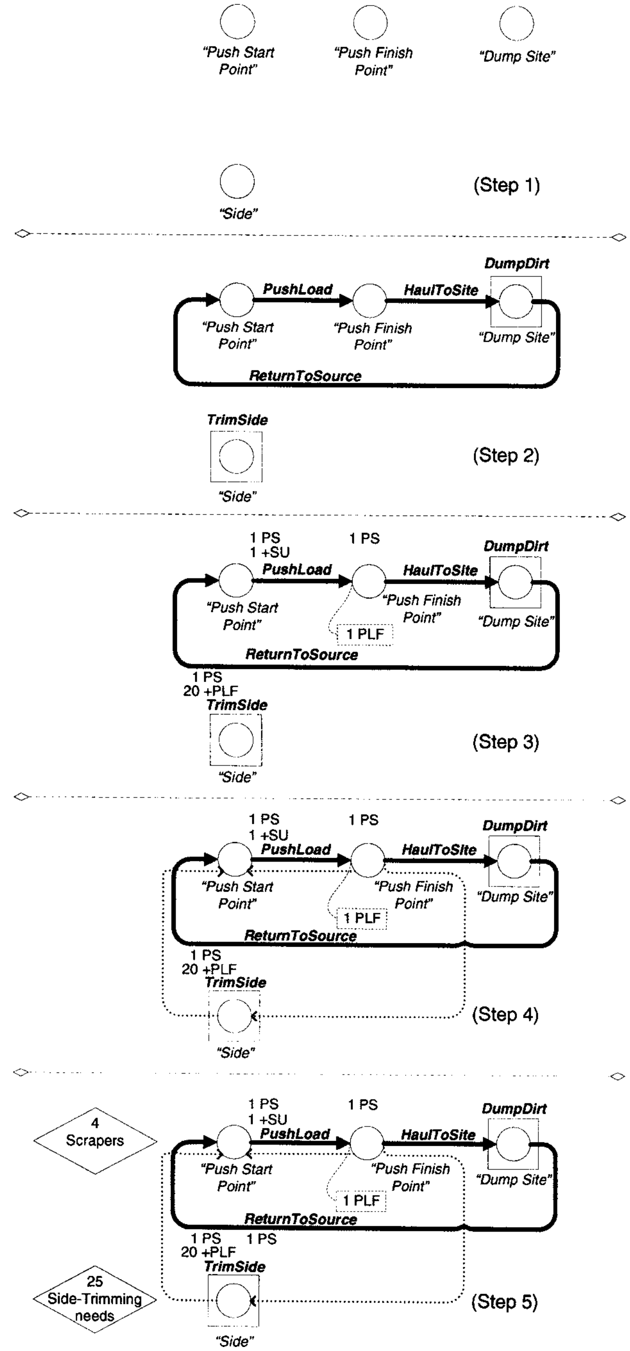

Fig. 5. General steps for process mapping and simulation modeling with SDESA

Circle the key locations in the site space that the scraper passes or stops by, including push start point, "push finish point," and "dump site" (shown in Fig. 5, step 1). In addition, the side-trimming workflow is associated with only one location, namely, the side.

2. Within each workflow, identify all activities and represent a production activity with a square node around its corresponding location circle [this is analogous to activity representation in the activity-on-node (AON) network diagramming

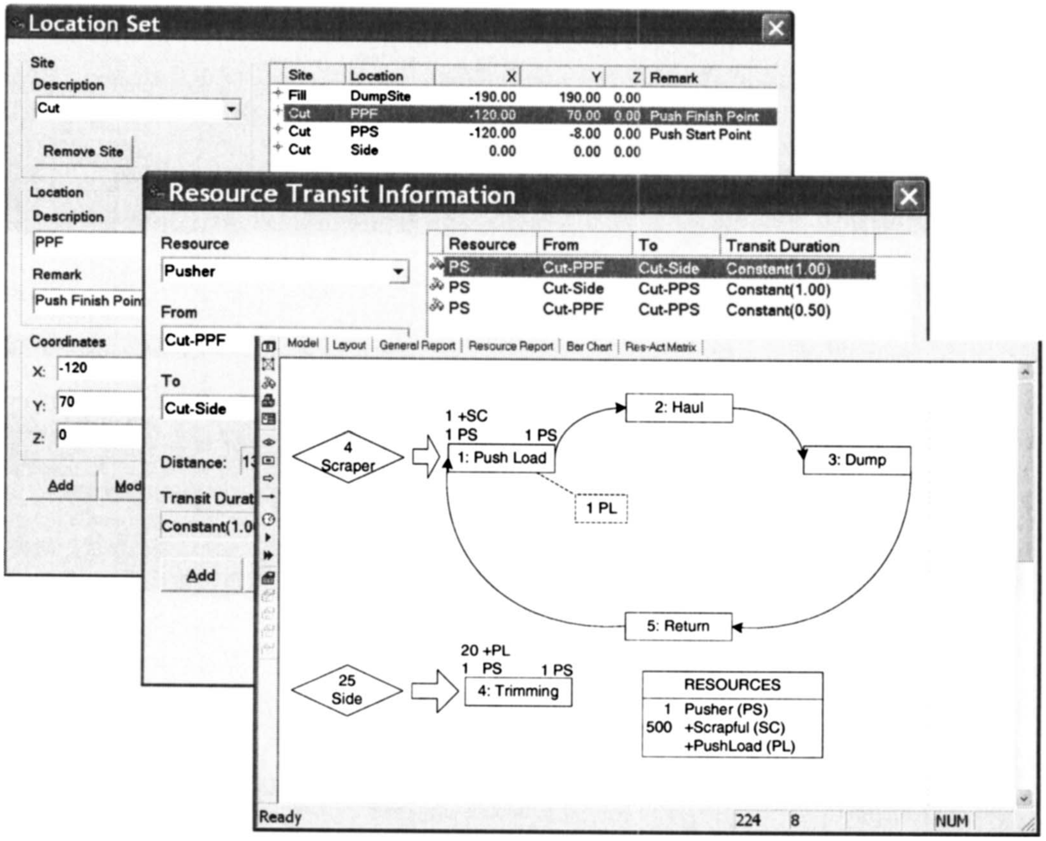

Fig. 6. Definitions of the location set, the resource transit-information system, and the network diagram in the SDESA computer system

technique]. Transit activity is denoted with an arrow linking two location circles denoting the origin and destination locations [this is reminiscent of the activity representation in the activity-on-arrow (AOA) network diagramming technique].

As shown in Fig. 5, step 2, in the scraper's workflow, three transit activities ("PushLoad," "HaulToSite," and "ReturnToSource") are represented with arrows linking two corresponding location circles; and one production activity "DumpDirt" is represented with a square node on the "dump site" location circle. A production activity, "TrimSide," is also marked on the side location.

3. Identify all the resources that need to be matched for executing each activity, including manpower and machinery (nondisposable resources) and material or information units (disposable resources, prefixed with "+" to be distinguishable from nondisposable resources).

The resource requirements for the push-load activity include one soil unit available and one pusher ready at the push-start-point location. At the end of push-load activity, it is specified in the model definition that the pusher will be released at the push-finish-point location; and one disposable RE called "push-load finished" (PLF) will be generated. As for the Trim-Side activity, the combination of 20 such PLFs and one pusher ready at the side location defines its resource needs (Fig. 5, step 3). As such, the two workflows are logically bound according to the technology and resource constraints given in the problem statement.

4. Define activity duration for each activity. In addition, specify resource transit times between various site locations in Res_TIS. In our case, the activity duration data are: 2 min for push-load, 4 min for HaulToSite, 1 min for DumpDirt, and 3 min for ReturnToSource. Res_TIS includes data entries for: (1) Pusher backtracking from push finish point to push start point (0.5 min), (2) pusher moving from push finish point to side (1 min), and (3) pusher moving from side to push start point (1 min). The three pusher transit routes are depicted as dashed arrows in Fig. 5, step 4.

5. Initialize the quantities and arrival times of flow entities associated with each workflow in a diamond block. Also initialize the type and quantity of REs available in the resource pool of the simulation model. Note both the initial values of available time and current location of each RE in the resource pool are assigned prior to the start of simulation. The two attributes of a RE reflect the current state of the system and are continuously traced and dynamically updated as simulation proceeds.

In Fig. 5, step 5, four scrapers are initialized in a diamond block, which is linked to the location circle "push start point." Also note, 25 flow entities, representing 25 side-trimming needs, are initialized at the "side" location. Meanwhile, one pusher and 500 soil units are initialized in the resource pool with their initial available time set as 0 and current location set as "push start point," respectively.

Converting the resultant process-mapping model (shown in Fig. 5, step 5) into a SDESA simulation model is straightforward (shown in the lower part of Fig. 6). The final SDESA model simply represents the production or transit activities in the process-mapping model with activity blocks. Those activity blocks are linked by arrows to form workflows. The location set, Res_TIS, and the network diagram of the model are given in Fig. 6. The computer simulation for the above case study has identified: (1) optimum number of scrapers to be four, to best match one pusher; (2) total duration of 1315 min; and (3) utilization

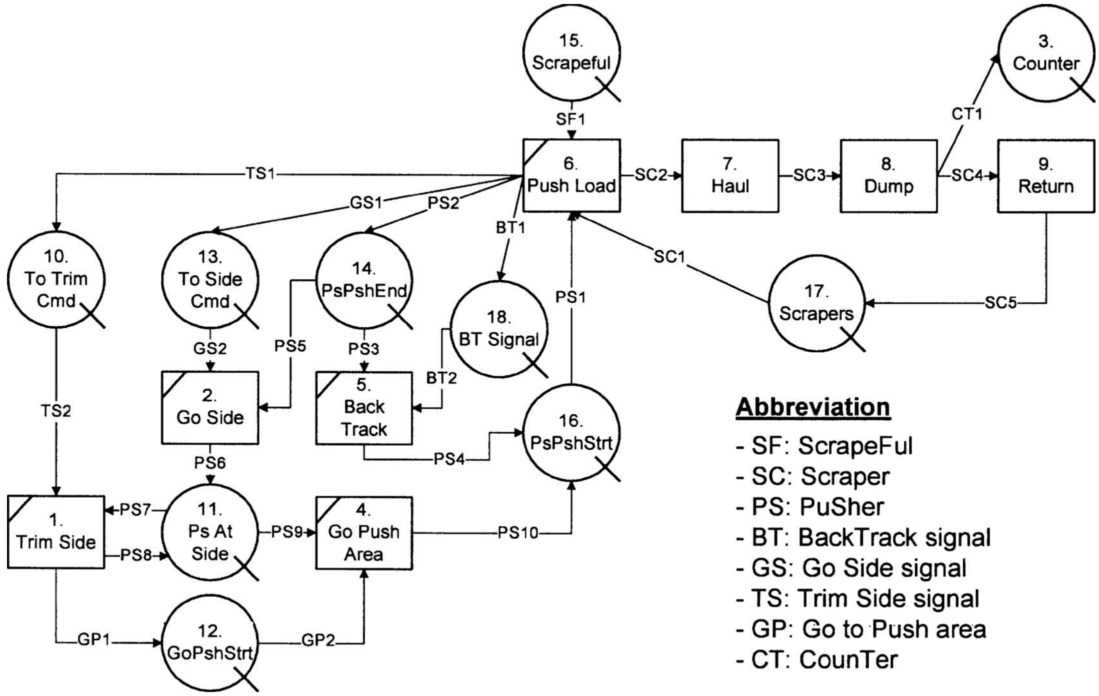

Fig. 7. ACD model for the earth-moving case in the STROBOSCOPE form

rates for the pusher and the scrapers being 100% and 95%, respectively.

STROBOSCOPE Model

In order to contrast SDESA with an established ACD approach, we also modeled the above earth-moving problem (identical definition of activity logic and times) with STROBOSCOPE. The activity cycle diagram model given in the STROBOSCOPE form is shown in Fig. 7. Note, STROBOSCOPE tags each arrow link in the ACD model to distinguish the simulation entities involved. For instance, in Fig. 7, the tag "SC" denotes scraper and "PS" implies pusher. The scraper earth-moving cycle, which is shown in the right portion of the ACD (Fig. 7), is straightforward to set up and understand. However, as shown in the left portion of the model, it becomes cumbersome and complicated to: (1) model the pusher's backtracking from the push finish point to the push start point and (2) realize the logic of moving the pusher to the side for side trimming once every 20 push-loads. It is also noted that in STROBOSCOPE, enabling the functions of consolidating or cloning simulation entities requires the modeler to write proprietary command code (e.g., "ENOUGH" and "DRAWAMT"). For example, the code of "ENOUGH GS2 ToSideCmd.CurCount>=20" means that 20 push-load completion signals will be accumulated before the que node "13. To Side Cmd" produces one simulation entity, GS2, which denotes a "Go to Side" signal. One simulation entity waiting at the que node "14. PsPshEnd" indicates the pusher resource is available for triggering the combi activity "2. Go Side," which models the pusher's transit from the push end point to side. Also, note the combi activity "5. Back Track" represents the pusher's backtracking from the "push finish point" to "push start point." Combi activity "4. Go Push Area" models the pusher's returning from side to push start point. In comparison with SDESA, the ACD model appears unwieldy for portraying the pusher's transit between various locations; as a result, forming the STROBOSCOPE model for our case-study problem obviously demands more time and effort than SDESA.

For cross-checking purposes, the STROBOSCOPE model was executed, resulting in the same outcome obtained from SDESA. To further verify the general applicability of the proposed simulation methodology for integrating operations modeling with site-layout modeling, the following section presents one case of modeling typical concreting operations in Hong Kong's building sites.

Hoist-and-Barrow Concreting Case

The "hoist-and-barrow" method is used to concrete a slab on the 20^{th} floor (56.4 m above ground) of a building in Hong Kong.

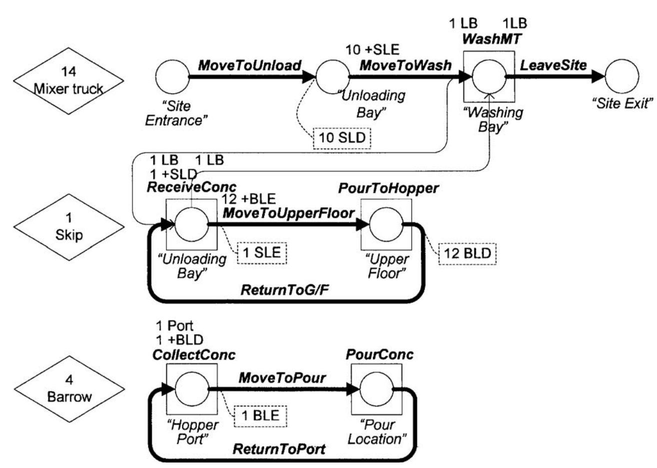

Fig. 8. Resultant process-mapping model for hoist-and-barrow concreting case

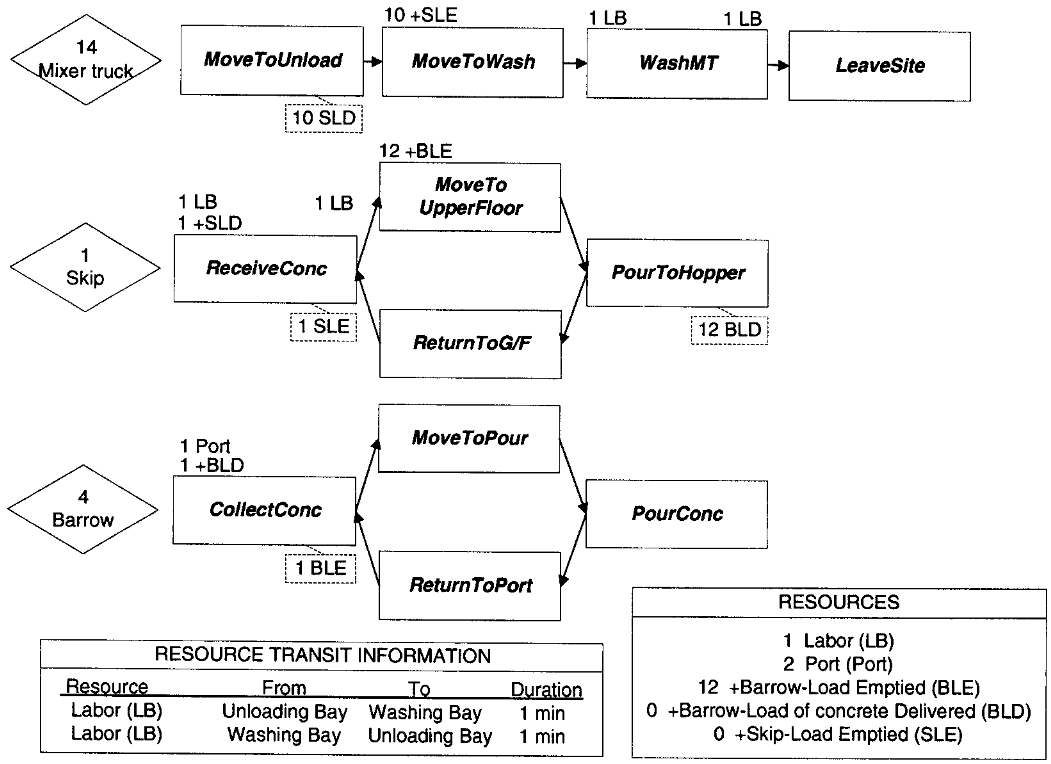

Fig. 9. Resulting SDESA simulation model for hoist-and-barrow concreting case

Fig. 8 shows the process-mapping model resulting from application of the modeling steps given in the previous section. It comprises three workflows: The mixer truck flow, the skip flow, and the wheelbarrow flow. The flow entities are initialized for each workflow in a diamond block, implying: (1) 14 concrete deliveries by mixer trucks; (2) one skip available to hoist concrete from the unloading bay at the ground level to the upper floor; and (3) four wheelbarrows used on the upper floor to pour concrete. It is noted that one truckload of concrete fills up the skip ten times; and one skip-load fills up a wheelbarrow 12 times.

The definition of disposable resources enforces the logic that binds together various workflows. For instance, 10 SLD, denoting "ten skip-loads of concrete delivered," are generated at the end of the "MoveToUnload" activity in the mixer-truck workflow; while one SLD is the resource requested to trigger the "ReceiveConcrete" activity in the skip workflow. One SLE, representing a signal of "one skip-load emptied," is produced as the result of completing the "ReceiveConcrete" activity; 10 SLEs combined will indicate one truck being emptied and define the resource requirements for the "MoveToWash" activity in the truck workflow. In a similar way, BLD, "one barrow-load of concrete delivered," and BLE, "one barrow-load emptied," logically relate the skip and barrow workflows. One BLE is generated at the end of "CollectConc" activity of the barrow workflow, and 12 BLEs combined trigger the start of "HoistToUpperFloor" activity of the skip workflow. In the actual situation, a laborer on the upper floor rang an electrical bell to request hoisting the next skip of concrete once the concrete container on the upper floor (a buffer between the skip and wheelbarrows) became empty.

In the current case, a laborer was observed to be involved in two activities at two different locations, namely, "ReceiveConc" at "Unloading Bay," and "WashTruck" at "Washing Bay." To accurately model the laborer's transit between serving two activities, 1 min transit time between the two locations is specified in the Res_TIS. In addition, there was limited space on the upper floor that permitted loading of, at most, two wheelbarrows at one time. So, two "port" resources, denoting the space-availability constraints, were added to the model as nondisposable resource entities; and one free port is part of the resource requirements for the CollectConc activity.

The resulting SDESA simulation model is shown in Fig. 9. By running computer simulation with activity-duration distributions based on site records, the following outputs were obtained: (1) total pour time of 7 h; and (2) the mixer trucks' inter-arrival time of 33 min, to achieve just-in-time concrete supply service. The simulation results found are a close match to the actual site performances.

Conclusions

This paper presents a new process-mapping and simulation methodology for modeling construction operations. Our research is focused on how to effectively model resource transit among various activity locations in the site system. We compared the modeling paradigms of AS and PI and exposed the difficulties and limitations of each in terms of modeling dynamic resource transit in a construction system.

In particular, we enhanced the algorithm and model structure for the SDESA (Lu 2003) into a generic process-mapping and simulation methodology by incorporating site operations and layout planning, while not sacrificing the simplicity of the original SDESA model. The process-mapping procedure is illustrated step by step using an earth-moving example featuring technology and resource constraints, resulting in generation of a process-mapping model, which describes the workflows over site locations. Converting the resultant process-mapping model into a SDESA simulation model is straightforward.

To contrast SDESA with the established ACD-based methodology, we also modeled the above earth-moving problem (using identical definitions of activity logic and times) with STROBOSCOPE. The STROBOSCOPE model was executed, resulting in the same outcome as obtained from SDESA. In comparison with SDESA, the ACD model appears unwieldy for portraying the pusher's transit between various locations. As a result, forming the STROBOSCOPE model for our case-study problem obviously demands more time and effort than SDESA. In addition, to further verify the general applicability of the proposed construction process-mapping and simulation methodology in practical settings, we present one case of modeling the typical concreting operations in Hong Kong's building sites.

To enable more sophisticated resource-allocation control, we have also added control variables, resource attributes, logic scripting, and iconic animation in implementing the proposed simulation methodology into a prototype computer system. Nevertheless, it is reemphasized that the most important step of applying simulation in construction is to develop a straightforward, systemic view of the real world, which will eventually lead to the generation of a simulation model able to capture workflows and resource flows over site locations in a construction system.

Acknowledgments

The writers are grateful to the anonymous reviewers of this journal for helping us improve the paper quality. The STROBOSCOPE software used in this research was downloaded from the website of Dr. Julio Martinez, Virginia Tech: http://strobos.ce.vt.edu/DownloadStrobo.htm. The work described in this paper was substantially funded by the Research Grants Council of the Hong Kong Special Administrative Region Government, China (Project No. PolyU 5049/02E).

References

Akinci, B., Fischen, M., Levitt, R., and Carlson, R. (2002). "Formalization and automation of time–space conflict analysis." *J. Comput. Civ. Eng.*, 16(2), 124–134.

Chau, K. W., Anson, M., and Zhang, J. P. (2004). "Four-dimensional visualization of construction scheduling and site utilization." *J. Constr. Eng. Manage.*, 130(4), 598–606.

Choo, H. J., and Tommelein, I. D. (1999). "Space scheduling using flow analysis." *Proc., 7th Annual Conf. of the Int. Group for Lean Construction*, Univ. of Calif., Berkeley, 299–311.

Chua, K. H. C., and Li, G. M. (2001). "Modeling construction operations with RISim." *J. Comput. Civ. Eng.*, 15(4), 320–328.

Evans, J. B. (1989). *Structures of discrete event simulation*, Wiley, New York.

Hajjar, D., and AbouRizk, S. M. (2000). "Application framework for development of simulation tools." *J. Comput. Civ. Eng.*, 14(3), 160–167.

Halpin, D. W. (1977). "CYCLONE—Method for modeling job site processes." *J. Constr. Div.*, 103(3), 489–499.

Halpin, D. W. (1989). *MicroCYCLONE user's manual*, Division of Construction Engineering and Management, Purdue Univ., West Lafayette, Ind.

Halpin, D. W. (1998). "Construction simulation: A status report." *Proc., 5th Canadian Construction Research Forum*, Univ. of Alberta and Alberta Construction Industry, Alberta, Canada, 33–41.

Hegazy, T., and Elbeltagi, E. (1999). "EvoSite: Evolution-based model for site layout planning." *J. Comput. Civ. Eng.*, 13(3), 198–206.

Hooper, J. W. (1986). "Strategy related characteristics of discrete-event languages and models." *Simulation*, 46(4), 153–159.

Huang, R. Y., and Halpin, D. W. (1993). "Dynamic interface simulation of construction operations." *Proc., 10th ISARC. Automation and Robotics in Construction*, G. H. Watson, R. L. Tucker, and J. K. Walters, eds., Elsevier, New York, 503–510.

Ioannou, P. G. (1988). "UM-Cyclone User's Guide." *Tech. Rep. No. UMCE89-12*, Dept. Civil Engineering, Univ. of Michigan, Ann Arbor, Mich.

Kalk, A. (1980). "INSIGHT: Interactive simulation of construction operations using graphical techniques." *Tech. Rep. No. 238*, Civil Engineering Dept., Stanford Univ., Stanford, Calif.

Koo, B., and Fischer, M. (2000). "Feasibility study of 4D CAD in commercial construction." *J. Constr. Eng. Manage.*, 126(4), 251–260.

Law, A., and Kelton, D. (2000). *Simulation modeling and analysis*, 3rd Ed., McGraw-Hill, New York.

Li, Z., Anson, M., and Li, G. (2001). "A procedure for quantitatively evaluating site layout alternatives." *Constr. Manage. Econom.*, 19, 459–467.

Lluch, J., and Halpin, D. W. (1982). "Construction operation and microcomputers." *J. Constr. Div.*, 108(1), 129–145.

Lu, M. (2003). "Simplified discrete-event simulation approach for construction simulation." *J. Constr. Eng. Manage.*, 129(5), 537–546.

Lu, M., and Wong, L. C. (2007). "Comparison of two simulation methodologies in modeling construction systems: Manufacturing-oriented PROMODEL vs. construction-oriented SDESA." *Autom. Constr.*, 16, 86–95.

Martinez, J. C. (1996). "STROBOSCOPE: State and resource based simulation of construction processes." Ph.D. thesis, Univ. of Michigan, Ann Arbor, Mich.

Martinez, J. C., and Ioannou, P. G. (1999). "General-purpose systems for effective construction simulation." *J. Constr. Eng. Manage.*, 125(4), 265–276.

Mawdesley, M. J., Al-jibouri, S. H., and Yang, H. (2002). "Genetic algorithms for construction site layout in project planning." *J. Constr. Eng. Manage.*, 128(5), 418–426.

Mohamed, Y., and AbouRizk, S. M. (2005). "Framework for building intelligent simulation models of construction operations." *J. Comput. Civ. Eng.*, 19(3), 277–291.

Oloufa, A. A., Ikeda, M., and Nguyen, T. (1998). "Resource-based simulation libraries for construction." *Autom. Constr.*, 7, 315–326.

Pidd, M. (1998). *Computer simulation in management science*, Wiley, New York.

Pritsker, A. (1986). *Introduction to simulation and SLAM II*, Wiley, New York.

Shi, J. J. (1999). "Activity-based construction (ABC) modeling and simulation method." *J. Constr. Eng. Manage.*, 125(5), 354–360.

Shi, J., and AbouRizk, S. M. (1997). "Resource-based modeling for construction simulation." *J. Constr. Eng. Manage.*, 123(1), 26–33.

Song, L., and AbouRizk, S. M. (2006). "Virtual shop model for experimental planning of steel fabrication projects." *J. Comput. Civ. Eng.*, 20(5), 308–316.

Thabet, W. Y., and Beliveau, Y. J. (1997). "SCaRC: Space-constrained resource-constrained scheduling system." *J. Comput. Civ. Eng.*, 11(1), 48–59.

Tommelein, I. D., Carr, R. I., and Odeh, A. M. (1994). "Assembly of simulation networks using designs, plans, and methods." *J. Constr. Eng. Manage.*, 120(4), 796–815.

Zhang, C., Hammad, A., Zayed, T. M., and Wainer, G. (2005). "Representation and analysis of spatial resources in construction simulation." *Proc., 2005 Winter Simulation Conf.*, WSC Foundation, Orland, Fla., 1541–1548.

Zhang, H., Shi, J. J., and Tam, C. M. (2002). "Iconic animation for activity-based construction simulation." *J. Comput. Civ. Eng.*, 16(3), 157–164.