Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining

Reza Akhavian, S.M.ASCE[1]; and Amir H. Behzadan, A.M.ASCE[2]

Abstract: In order to develop a realistic simulation model, it is critical to provide the model with factual input data based on the interactions and events that take place between real entities. However, the existing trend in simulation of construction fleet activities is based on estimating input parameters such as activity durations using expert judgments and assumptions. Not only may such estimations not be precise, but project dynamics can influence model parameters beyond expectation. Therefore, the simulation model may not be a proper and reliable representation of the real engineering system. In order to alleviate these issues and improve the current practice of construction simulation, a thorough approach is needed that enables the integration of field data into simulation modeling and systematic refinement of the resulting models. This paper describes the latest efforts by authors to design and test a novel methodology for multimodal process data capturing, fusion, and mining that provides a solid basis for automated generation and refinement of simulation models that realistically represent construction fleet operations. Different modes of operational data are collected and fused to facilitate the discovery of operational knowledge required to create realistic simulation models. The developed algorithms are validated using laboratory scale experiments and analytical results are also provided. The main contribution of this research to the body of knowledge is that it lays the foundation to systematically investigate whether it is possible to robustly discover computer-interpretable knowledge patterns from heterogeneous field data in order to create or refine realistic simulation models from complex, unstructured, and evolving operations such as heavy construction and infrastructure projects. **DOI: 10.1061/(ASCE)CO.1943-7862.0000775.**

CE Database subject headings: Construction management; Simulation; Data collection; Information management.

Author keywords: Construction; Simulation; Data driven; Knowledge discovery; Data fusion; Data mining; Heavy equipment; Real time; Earthmoving.

Introduction

Within the construction engineering domain, the use of discrete event simulation (DES) to model resource interactions and operational logic has been the subject of several studies (Hajjar and AbouRizk 1999; Martinez and Ioannou 1999; Lu 2003). A DES model is an event-based representation of project activities that constitute an engineering system. Considering factors such as complexity and scale and given the multidisciplinary nature of a construction or infrastructure project, decision makers and field engineers may rely on computer-generated simulation results to study key performance indicators including resource allocation, equipment utilization, site planning, and bottleneck elimination (AbouRizk and Shi 1994). Depending on information availability at different project stages, the level of detail and the scope of resulting simulation models may vary. The granularity and credibility of results generated by these simulation models is a critical factor in determining whether such models can be readily used by project decision makers (Banks 1998).

An extensive literature review conducted by the authors revealed that most construction simulation models are mainly used during the early planning and design because they are built on rigid assumptions and design parameters (e.g., precedence logic, activity durations). For instance, prior to launching a DES model, one has to carefully identify different activities and diligently create an activity cycle diagram (ACD) that prescribes the flow of resources. All modifications to this logic as a result of changes in the real system must be manually done, which may prove to be a tedious task if not impossible. Similarly, activity durations must be provided prior to running the model. Moreover, there is often no systematic way to refine the simulation by overwriting its variables based on the actual conditions on the ground. In all such cases and in the absence of an inclusive methodology to incorporate real field data as the construction evolves, modelers rely on data from previous projects, expert judgments, and subjective assumptions to generate simulations that can reasonably predict future performance. These and similar shortcomings have to a large extent limited the use of traditional DES tools to preliminary studies and long-term planning of construction projects. A longstanding research challenge is how to generate simulation models that are responsive to real-time changes in the project during the execution (construction) phase. What makes this fundamental question of utmost importance is that the construction phase of many engineering projects may in one way or the other be affected by uncertainties such as weather delays, safety incidents, unforeseen site conditions, and equipment breakdowns that are not easy to mathematically formulate ahead of time and predict before commencing the actual project. In fact, previous studies have indicated that on average, construction schedules and plans experience changes up to 70% (Daneshgari and Moore 2009). Therefore, proper simulation-based

[1]Ph.D. Student, Dept. of Civil, Environmental, and Construction Engineering, Univ. of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816-2450. E-mail: reza@knights.ucf.edu

[2]Wharton Smith Faculty Fellow and Assistant Professor, Dept. of Civil, Environmental, and Construction Engineering, Univ. of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816-2450 (corresponding author). E-mail: amir.behzadan@ucf.edu

Note. This manuscript was submitted on January 31, 2013; approved on July 2, 2013; published online on August 5, 2013. Discussion period open until January 5, 2014; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Construction Engineering and Management***, © ASCE, ISSN 0733-9364/04013021(11)/$25.00.

operations-level planning and control during project execution require that attributes of the corresponding simulation model elements are modified with progress of the project so that ultimately the simulation model can be completely adaptable and responsive to the latest site conditions. This underlines the importance of a robust methodology that supports the prospect of reliable collection and processing of field data and effective extraction of relevant contextual knowledge. Recent advancements in field data acquisition and remote sensing technologies alleviate the challenge (i.e., required time and cost) of manual data collection. For instance, researchers have recently explored different sensor technologies for material tracking (Caldas et al. 2006; Jang and Skibniewski 2007), human motion tracking (Han et al. 2011), labor and equipment tracking (Navon 2005; Behzadan et al. 2008; Akhavian and Behzadan 2011), and vision-based detection and tracking (Brilakis et al. 2011). However, the majority of previous research has targeted certain project tasks such as controlling delivery and receipt of construction materials, enhancing safety, progress monitoring, and productivity assessment. In light of this, research is still needed to design generic and robust data sensing and handling strategies that yield the maximum amount of information from the minimum volume of data (MacKay 1992; Ling 2011). Without transforming raw data to process information and ultimately contextual knowledge, collecting large volumes of sensor-based data can provide little value for project planning and optimization.

Main Contributions to the Body of Knowledge

In the past, the problem of transforming raw process data into contextual knowledge and using the extracted knowledge to automatically generate simulation models has been the subject of some studies within manufacturing and industrial engineering domains (Yuan et al. 1993; Son and Wysk 2001; Véjar and Charpentier 2012). Unlike manufacturing where the production environment is fully controlled and ambient factors are kept to a minimum, construction projects take place in unstructured environments that are hard to comprehend and formulate ahead of time. Thus, simulation modelers often tend to use simplifications, assumptions, and prescriptive parameters to build construction simulation models. Although this approach may streamline the modeling process, it may as well take away from the flexibility and extensibility of the model, negatively impact its accuracy in representing the project dynamics, and ultimately be detrimental to the model reliability, verification, and validation (Davis 1998). Hence, the main contribution of this research to the body of knowledge is that it lays the foundation to systematically overcome these challenges by exploring whether it is possible to create and refine realistic simulation models from complex, unstructured, and evolving operations such as heavy construction and infrastructure projects. This will be achieved by introducing a framework capable of automatically generating and updating simulation models based on the latest field data collected using a ubiquitous distributed sensor network mounted on a construction fleet. These heterogeneous data sets are fused into a reasoning process that extracts contextual knowledge necessary to generate or refine simulation models. The generated simulation model is constantly updated using new incoming data streams. Akhavian and Behzadan (2012) investigated the applicability of dynamic data-driven application simulation (DDDAS) to update existing DES models of construction operations using single-modal data. The material in this paper builds on this previous work by (1) establishing a framework for extracting contextual knowledge from raw streaming multimodal field data and (2) eliminating the burden of manually creating and continuously updating simulation models by enabling automated generation of realistic models from evolving engineering systems.

Methodology

In this section, the key components of the designed methodology are discussed and the current state of knowledge is presented to highlight the main departure points of the presented research.

Multimodal Data Collection and Fusion

Human brain justification is the major tool and the best example of data fusion in action (Razavi and Haas 2012) in traditional simulation paradigms to determine required parameters and variables. Hence, what a modeler's brain does in defining key parameters of a simulation model is, admittedly, fusion of heterogeneous data from multiple sources including previous project databases, engineering judgment, site layout parameters, and effect of ambient factors. The basic concepts and existing techniques of multimodal data acquisition and fusion have been investigated in several research studies thst aimed at introducing solutions to specific problems within construction engineering. For example, Kannan and Vorster (2000) explored developing an experience database to fuse payload, temperature, and cycle-time data for the load activity in an earthmoving operation. However, haul and return activities were not included in their work because positional data were not collected. Moreover, because data were collected using dump trucks' preinstalled on-board instrumentation (OBI), no additional information describing, for example, the interaction between dump trucks and loaders and operational logic were provided. In another study, as-design spatial information was fused with as-is laser scanner spatial data to detect construction defects (Akinci et al. 2006). Researchers also worked with positional data from global positioning system (GPS) and radio frequency identification (RFID) to estimate the coordinates of construction equipment and inventory items (Grau and Caldas 2009). More recently, spatial (e.g., soil type) and temporal (e.g., weather) data were fused to support construction productivity monitoring (Pradhan and Akinci 2012). Although all such studies explored data fusion techniques within the construction engineering domain, none investigated a systematic method to collect and synchronize data from multiple resources of different types in order to discover knowledge about the ongoing activities (e.g., state of resources, operational logic) and reveal potential patterns existing in constantly streaming data streams.

The main focus of this paper is to demonstrate the suitability and reliability of multimodal process data collected from active fleet in heavy construction projects (e.g., road construction, pavement resurfacing, earthmoving, mining) in generating and refining realistic DES models describing such operations. In the developed methodology, three modes of heterogeneous process data (i.e., position, orientation, and weight) collected from a distributed network of sensors are synchronized to determine the state of resources (i.e., equipment) that are involved in various stages of an arbitrary operation. The extracted knowledge will be used to generate and update a simulation model corresponding to the real engineering system. As described in the following sections, what distinguishes the presented data collection and handling framework from existing methods is that in order to initiate the reasoning process, minimum (if any) prior knowledge about the existing site layout and location and configuration of different resources is required. The developed technique is capable of intelligently observing (sensing) the real system and accordingly building (generating)

or refining a computer model that best describes the dynamics and evolving nature of the ongoing operations. Once the data collection process is initiated, incoming data streams from individual sensors are captured and analyzed to the extent that a sufficient level of operational knowledge about the ongoing processes can be secured. This initiation stage is an essential component of the framework. The goal is to train the system with the minimum amount of incoming raw data necessary to sufficiently describe (with good confidence) the nature and logic of ongoing site activities. As previously discussed, unlike existing methods that use collected data for localization or context awareness, in this paper data are used to provide information and knowledge necessary to generate a realistic simulation model considering that the level of detail and amount of collected data are subject to change as the project evolves. Hence, computational efficiency and cost are among major factors in selecting the most appropriate data collection strategy that can be sustained for the life cycle of the project. Very often, captured data are of tremendously large volume and contain a high noise ratio (Razavi and Haas 2012), such that data cleaning and analysis takes a long time. In the presented research, the goal is to use almost all collected data one way or another in the reasoning process, and consequently keep the noise and data redundancy to a minimum. Existing videotaping (i.e., vision-based) techniques, as an alternative, may prove to be computationally inefficient for the purpose of this research because the level of detail and volume of unnecessary collected data (in each video frame) may easily exceed the computational efficiency requirements. For instance, while in the presented work, all collected data are directly relevant to the knowledge-extraction process, in vision-based techniques, a large amount of irrelevant data (e.g., background scene) is inevitably collected, which may not contribute to the intended purpose of the data collection task. In particular, most computer vision techniques model images as two-dimensional (2D) arrays of intensity values (i.e., gray levels from 0 or black to 255 or white), which implies that a 1-s video of 30 frames per second with a fair resolution of 640 × 480 will contain 9,216,000 integer numbers (32 bits). Processing this volume of data will require intensive computational effort especially when real-time or near-real-time response is of essence. Also, vision-based techniques such as background subtraction fail to distinguish between object types, thus making it difficult to detect target equipment (Park et al. 2012). The developed algorithm in this research is able to identify different states of construction equipment regardless of the pose they hold, visual occlusion, or environmental conditions (e.g., illumination), all of which have been in fact identified as major research areas in vision-based tracking (Gong and Caldas 2010; Yang et al. 2010; Rezazadeh Azar and McCabe 2012). Hence, the data collection strategy developed in this research was mainly motivated by the need for a reliable and ubiquitous method easily deployable to collect relevant data in minimum time and with the least computational cost. In the developed methodology, three classes of sensing devices were used: a network of ultra-wideband (UWB) receivers and tags to track resource positions, attitude and heading reference system (AHRS) to track resource articulation, and Zigbee-enabled weight sensors to track the amount of transported material.

Knowledge Discovery and Reasoning Process

The core of the developed framework is the ability of extracting meaningful (contextual) knowledge necessary to automatically generate and refine a simulation model that adequately describes the real system. The need for a solid methodology to extract useful and relevant knowledge from a large amount of collected data has been highlighted in the past by several researchers in construction and civil engineering domains (Soibelman and Kim 2002; Chen et al. 2005; Pradhan and Akinci 2012). Efficient knowledge extraction requires that relevant data sets are identified and irrelevant and redundant data are eliminated. Soibelman and Kim (2002) indicated that the ability to conduct valid data analysis and useful knowledge discovery depends on the availability of clean relevant data. Similarly, the importance of quality assessment in data extraction during knowledge discovery is emphasized in the literature (Chen et al. 2005). Hence, an efficient knowledge extraction mechanism should (1) employ methods that run on the least possible amount of data, and (2) extract reliable contextual knowledge from relevant data sets.

The first step to extract basic operational knowledge about a resource is to detect the state of that resource. In general, the overall state of a resource can be described using a binary classification of idle or busy. However, as far as operations-level simulation of construction activities is concerned, this classification is too granular and may cause confusion or misunderstanding. Fig. 1 shows simple equipment taxonomy in a typical earthmoving operation. As shown in this figure, knowing that a truck is idle (not moving) (in the absence of any other data), one may easily conclude that it is out of service (e.g., has a flat tire) and needs mechanical maintenance, whereas another logical conclusion could be that it is being loaded by a loader and thus is not moving. Therefore, at the operations level, this high-level classification should be further broken down

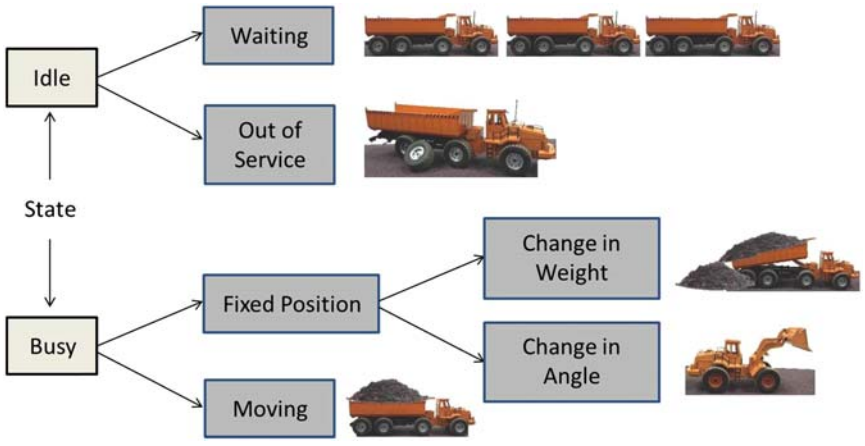

Fig. 1. Taxonomy-based state classification of construction fleet (images by authors)

into more meaningful subcategories. Fig. 1 also shows subcategories for the busy state in which if certain physical motions are observed, the resource state can be categorized as busy.

Identifying the correct state of a resource is critical to properly describing ongoing activities in an engineering system. In essence, activities consume resources and their start and end events correspond to when resources are drawn or released by them (Martinez and Ioannou 1994). The reasoning algorithm observes the trend of data that corresponds to each resource state and tries to discover the knowledge (e.g., activity start and end events, duration, resource levels) required to describe project activities. Most often, multiple modes of streaming heterogeneous (i.e., diverse in nature, content, and format) data may need to be evaluated to determine the true state of a resource. Once the start and end events of an activity are determined, activity durations can be calculated by comparing the time stamps of these events. However, because incoming data are often heterogeneous, they have to be first fused and transformed into a common temporal system before any contextual knowledge can be extracted. Another important issue that must be considered is the overall site layout and the approximate locations of where resources are idle (waiting to be drawn by activities) or busy (already drawn by activities). Generally, intensity of fleet position data can assist in this regard. For example, streaming positional data transmitted from a dump truck shows a higher intensity in waiting, loading, or dumping zones (where it is idle the most) compared to hauling routes (where it is moving). Clustering these intensity data helps determine the boundaries of regions that represent waiting queue, dumping or loading areas, and hauling routes. In a dynamic environment such as a highway project, where the location of work zones and routes change, models that rely on a fixed layout (Lee et al. 2008; Andoh et al. 2012) may not result in a realistic output. Using an intensity-clustering technique, however, changes in the site layout can be constantly monitored to allow simulation model variables (e.g., haul distances, activity durations) to be accordingly updated.

k-Means Clustering

Clustering methods are used in knowledge discovery and data mining (Fayyad et al. 1996), pattern recognition and pattern classification (Duda et al. 1973), and machine learning (Bishop 2006). There are two major types of clustering algorithms: hierarchical and partitioning methods (Berkhin 2002). In this research, clustering is used to find work zones in a jobsite where equipment spends most of its time (e.g., loading and dumping areas). As a result, hierarchical methods that produce a set of nested clusters similar to a hierarchical tree are not applicable. Partitioning methods, however, divide data points into nonoverlapping clusters such that each point belongs to exactly one cluster (Ali et al. 1998) and therefore they are suitable to detect distinct areas with a high population of positional data points. Probabilistic clustering, k-medoids, and k-means are the three subsets of partitioning algorithms. The first two methods are computationally complex and are predominantly used in pattern recognition applications (Mirkin 2005). Moreover, in k-medoids, centroids must be exactly at the center of clusters, whereas in k-means they can be anywhere in the sample space (which is a reasonable condition in a cluster of positional data points). Therefore, k-means, which is an iterative two-step algorithm (Berkhin 2002), was used in this research. As stated in Eq. (1), the goal of the k-means algorithm is to minimize the sum of squared error for each cluster:

$$J(C) = \sum_{i=1}^{k} \sum_{x_j \in c_i} (x_j - m_i)^2 \tag{1}$$

where x_j = individual point within the cluster C_i; m_i = mean; and the goal is to minimize score function J for all clusters. Initially, given a set of k-means $[m_1^{(1)}, m_2^{(1)}, \ldots, m_k^{(1)}]$, the algorithm partitions n data points into k clusters by assigning each data point to its nearest centroid. As stated in Eq. (2), x_p belongs to $C_i^{(t)}$ if it is closer to m_i^t than it is to m_j^t. This is shown in Eq. (2) where $C_i^{(t)}$ is the cluster i in the tth iteration, x_p is data point x, and m_i^t and m_j^t are centroids of clusters i and j, respectively.

$$C_i^{(t)} = \{x_p : (x_p - m_i^t) \le (x_p - m_j^t) \quad \forall\ 1 \le j \le k\} \tag{2}$$

Then, the new centers are computed to match the sample means of their assigned data points, as calculated using Eq. (3):

$$m_i^{t+1} = \frac{1}{|C_i^{(t)}|} \sum_{x_j \in c_i} x_j \tag{3}$$

The iterative process of assigning data points and readjusting means continues until it converges to a steady state and eventually stabilizes.

Contextual Knowledge Discovery Using k-Means

Using the k-means clustering method and provided with prior knowledge indicating the expected number of work zones (e.g., loading area, dumping area, resource waiting zones), it is possible to partition multimodal streaming data based on their intensity. If more than two modes of data are captured, k-means can be also applied on n-dimensional ($n > 2$) vector data to identify k clusters in the n-dimensional space. In the presented methodology, weight data constitute the third dimension in addition to the xy coordinates of each point, and thus k-means is applied to position-weight data points located inside a three-dimensional (3D) xyw space. Therefore, in finding the locations of designated work areas, weight serves as an important attribute of each point and helps in identifying states of equipment that are located in neighboring (and sometimes, spatially close) zones but have different loading conditions (e.g., loading area where weight is increasing versus loading queue where weight is constant and close to a minimum value).

As a motivating case, consider a simple earthmoving operation in which a front-end loader is tasked with loading a dump truck. Therefore, there are two locations where the intensity of points is relatively high, and as a result there will be two clusters: loading area and dumping area. Logically, what connects these two clusters are the routes: the haul route connects the loading area to the dumping area, and the return route connects the dumping area back to the loading area. Because the xyw data set is populated using positional and weight data captured from each work cycle, k-means dynamically calculates the approximate centroids of the two clusters. These clusters are then marked as representing the loading area (marked with abrupt weight increase) and dumping area (marked with abrupt weight decrease). Thus, at least two modes of data (i.e., position and weight) are needed to properly identify active loading and dumping zones. Clearly, not all earthmoving operations are as simple as the one described, and often multiple dump trucks are employed to haul soil, which may cause waiting queues to form near the loading and dumping areas, thus creating other intense clusters. Moreover, there may be rare situations in which dump trucks remain idle in an arbitrary location other than the loading area, dumping area, loading queue, or dumping queue. This place can be a service area where equipment receives periodic or random maintenance. Thus, the data reasoning process should be inclusive to cover all special cases that are likely to occur.

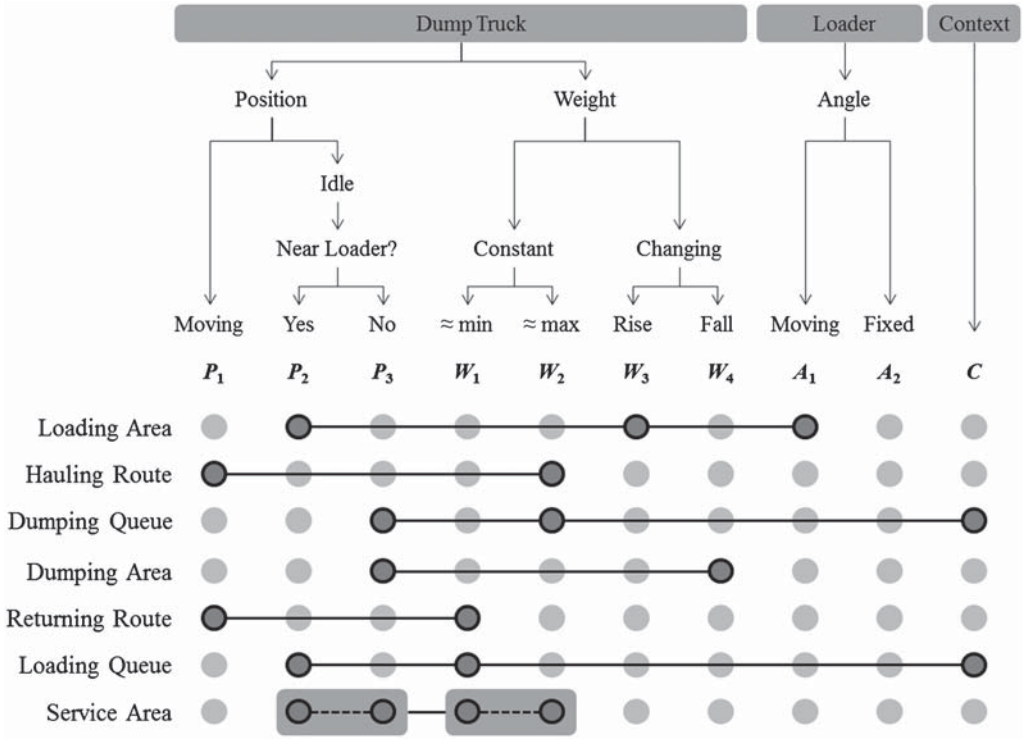

Fig. 2. Taxonomy of dump truck activities in an earthmoving operation based on multimodal process data and operational context

As such, a robust complex algorithm is needed to extract proper contextual knowledge. Fig. 2 depicts various combinations of data modes and trends that result in different states of a dump truck in an earthmoving operation.

In Fig. 2, process data and operational context are linked using solid lines (that represent logical AND) or dashed lines (that represent logical OR). For instance, a dump truck is loaded in the loading area if it is idle in the loader's proximity, its weight is increasing, and when the loader is working (represented by a changing boom angle). This can be represented as $P_2 \bigcap W_3 \bigcap A_1$. Likewise, if a dump truck is moving and its weight is close to its maximum value ($P_1 \bigcap W_2$), it can be concluded that the dump truck is traveling on the hauling route. In the dumping area, the dump truck is not moving while its weight is decreasing ($P_2 \bigcap W_4$). Once the soil is dumped, the dump truck travels back on the returning route, which requires the dump truck to be moving while its weight is close to a minimum value ($P_1 \bigcap W_1$). This taxonomy provides satisfactory results as far as major activities (load, haul, dump, and return) in an earthmoving cycle are concerned. However, in an operation in which multiple pieces of equipment are used, it is very likely that at certain times dump trucks have to wait in queues before they are drawn to activities. For instance, if a loader is already serving a dump truck, a second dump truck has to wait in a loading queue before the loader becomes available again. Likewise, if the dumping area can only accept one dump truck at a time, all other dump trucks that arrive to the vicinity of the dumping area need to first wait in a dumping queue. As stated previously, another example of a location where dump trucks may remain idle is the service area. If no prior information is provided as to where the service area is located (e.g., somewhere along the hauling route or along the returning route), then the reasoning process must consider all data and context combinations that may correspond to a dump truck inside the service area. For instance, while one may assume that a stationary dump truck that has a weight close to a maximum value is in a dumping queue waiting to dump its load, in reality, that same dump truck may be idle inside a service area that is located somewhere along the hauling route. To resolve these confusing cases, additional contextual information is needed. A dump truck waits inside a dumping queue only if the dumping area is occupied by other dump truck(s). Hence, what distinguishes a dumping queue from a service area is that from the moment a dump truck enters a dumping queue until it leaves the queue, the dumping area is continuously occupied. Using a similar logic, what distinguishes a loading queue from a service area is that from the moment a dump truck enters a loading queue until it leaves the queue, the loading area is continuously occupied. All other cases in which a dump truck is not moving and its weight is constant are considered instances of maintenance or repair work occurring inside the service area, represented by $(P_2 \bigcup P_3) \bigcap (W_1 \bigcup W_2)$.

Finally, it should be noted that although this reasoning process covers the majority of scenarios involving a dump truck, there are always specific (and rare) cases that may not be detected as expected. However, compared to the majority of activity and queue instances that are correctly identified, the effect of such exceptions on the overall performance of the reasoning process and clustering algorithm is minimal. In any case, if exceptions become rules (i.e., statistically significant), they can be systematically detected and represented as recurring events.

Automated Simulation Model Generation

As stated previously, a major contribution of this research to the body of knowledge is that it will ultimately provide means and methods necessary to conduct (near-)real-time operations-level planning, look-ahead scheduling, and short-term decision making by enabling the automated generation of adaptive simulation models using the contextual knowledge extracted from multimodal

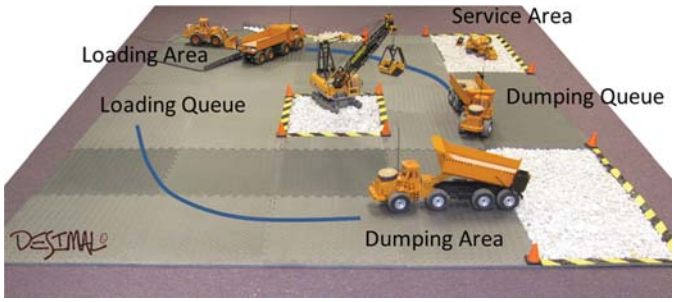

Fig. 3. Layout of laboratory experiments (image by authors)

Table 1. Specifications of Employed Sensors in the Distributed Network

Sensor type	Specifications	
Load Cell	Capacity	5, 10, or 20 kg
	Accuracy	±0.02%
	Resolution	24 bits
	Update rate	16 Hz
UWB	Accuracy	15 cm in 3D real time
	Update rate	0.00225 Hz up to 33.75 Hz (16 Hz in the experiments)
	Radio frequencies	Ultra-wideband 6–8 GHz
AHRS	Roll and pitch accuracy	0.8° RMS
	Heading accuracy	0.5° RMS
	Resolution	<0.5°

data sets. A simulation model generator is a tool for translating real system logic to simulation language, thus enabling a computer to represent the behavior of the model (Mathewson 1984). Previous efforts on this topic have been mainly limited to manufacturing and industrial engineering where product trajectories in a structured network of modules were used to generate adaptive manufacturing simulations (Véjar and Charpentier 2012). In another example, an automated simulation model generator was developed by Son and Wysk (2001) for real-time shop floor control. Yuan et al. (1993) developed a DES generator for operational systems with applications in manufacturing activities such as fabrication, machine setup, assembly, and part transportation. Using an input file of natural language (NL) components (e.g., electronics assembly words, expressions, and expectations), Ford and Schroer (1987) developed the electronic manufacturing simulation system (EMSS). The earliest use of NL interface, however, was the NL programming for queuing simulations (Heidorn 1974).

Unlike manufacturing and industrial systems in which the environment is fully controlled and structured, in many construction projects, resource and operational dynamics and the presence of ambient factors can intensify uncertainties. Thus, if simulation models are not linked to field data, they will soon become outdated. Despite this, there have been few previous attempts within the construction engineering domain to establish a systematic solution to this problem. For example, a look-ahead scheduling for heavy construction projects was described by Song et al. (2009) in which real time GPS data were used to update a simulation model. In another example, Bayesian updating of input variables of a simulation model was suggested for a tunneling project where the penetration rate of a tunnel-boring machine (TBM) was continuously obtained and used to update a distribution function to estimate completion time (Chung et al. 2006). However, to the best of authors' knowledge, there has been no systematic research within this domain to evaluate the potential benefits of this subject. In this research, a construction simulation model generator plays a key role because it receives and combines user input (e.g., number and types of resources, project type) and extracted operational knowledge (e.g., activity durations, site layout).

Results

In order to validate the designed methodology, several laboratory experiments were conducted on a test bed, which consisted of a model jobsite and remotely controlled equipment models (Fig. 3).

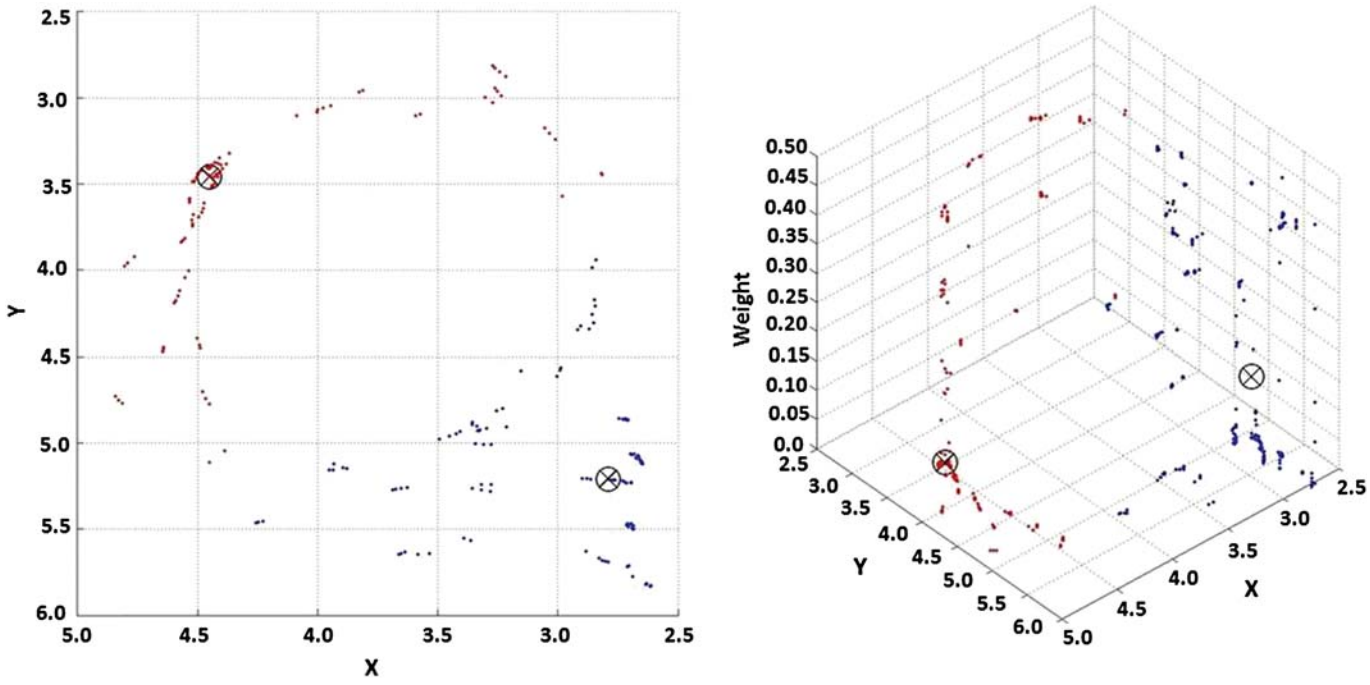

Fig. 4. Results of *k*-means clustering algorithm applied to the positional and weight data in 2D (*xy*) and 3D (*xyw*) spaces for Experiment 1

J. Constr. Eng. Manage.

Table 2. Centroids of the Detected Clusters Using *k*-Means for Experiment 1

Identified cluster	*x* (m)	*y* (m)	*w* (g)
Cluster 1	4.45050	3.46007	0.000400
Cluster 2	2.79147	5.20749	0.121125

Table 3. Observed versus Extracted Activity Duration Means and Standard Deviations for Experiment 1

	Observed duration (s)		Extracted duration (s)	
Activity	Mean	Standard deviation	Mean	Standard deviation
Load	13.6	8.4	15.4	9.18
Haul	33.8	5.9	34.2	4.92
Dump	11.5	3.4	9.80	1.83
Return	30.4	2.9	32.0	9.49

Equipment positions were captured by an UWB network, loader boom angle was sensed by an AHRS tracker, and Zigbee-enabled sensors tracked the weight of material transported by dump trucks. Table 1 shows basic specifications of these sensors. As listed in Table 1, the accuracy of the UWB sensors is 15 cm in a 3D space (and much better in a 2D space). Therefore, considering the update rate of 16 Hz used when conducting experiments in this research, and also given the 12-m^2 test bed that provided a distance of approximately 5 m between loading and dumping areas, the accuracy of the sensor was deemed acceptable. Two such experiments are detailed subsequently and results are provided.

Experiment 1: One Loader and One Dump Truck, without Service Area

In this experiment, the simplest case of an earthmoving operation including one front-end loader and one dump truck was modeled. In each cycle, the loader put soil in the dump truck (inside the loading area), the dump truck hauled the load, dumped it in a designated dumping area, and returned to the loading area to start the next cycle. Hence, there were no waiting queues. It was also assumed that the dump truck would not stop anywhere else on its path due to maintenance-related events. Thus, the *k*-means algorithm was applied with $k = 2$. In addition, the reasoning algorithm captured the trend of weight data and identified loading and dumping clusters. Fig. 4 illustrates the plots of collected positional and weight data in 2D (*xy*) and 3D (*xyw*) spaces. In the 3D plot, the vertical axis shows weight values. Hence, in the vicinity of the loading area, 3D points show a rising trend (increase in weight) while in the vicinity of the dumping area, 3D points show a falling trend (decrease in weight). The developed *k*-means algorithm successfully detected two clusters and found the centroids as shown in Table 2.

Considering the load data trend, it was found that clusters 1 and 2 corresponded to the loading area and dumping area, respectively. The developed reasoning technique provided further information about the state of the dump truck, which in turn helped determine activity durations in each cycle. Statistical analysis on pools of activity durations provided the mean and standard deviation of each activity duration. Table 3 compares observed values (from experiment video) with extracted values (using the reasoning process) of activity durations. In order to assess if the mean of activity durations obtained from the developed methodology is in good agreement with the observed values, Student's t-test was used (Sall et al. 2012). Student's t-test is a popular statistical analysis used to compare means of different populations. In a nutshell, the Student's t-test investigates whether there is a statistically significant difference between the mean values. According to the results, the t-statistic for load, haul, dump, and return activities are 0.32, 0.11, 0.98, and 0.36, respectively, and the *p*-values of all of them are greater than 0.05 (i.e., confidence level of 95%). This indicates that in the first experiment, the means of observed and extracted durations are not statistically significantly different.

By capturing sensor data streams and using proper stream-mining techniques, the developed methodology can also determine possible changes in the locations of work areas. Road construction is a good example because the cut and fill zones constantly change. Thus, Experiment 1 was further evolved by making the dump truck dump soil on different spots. As shown in Fig. 5, while the location of the loading area was almost unchanged, the centroid of the dumping area moved with time. This change in the layout was captured as more field data were collected.

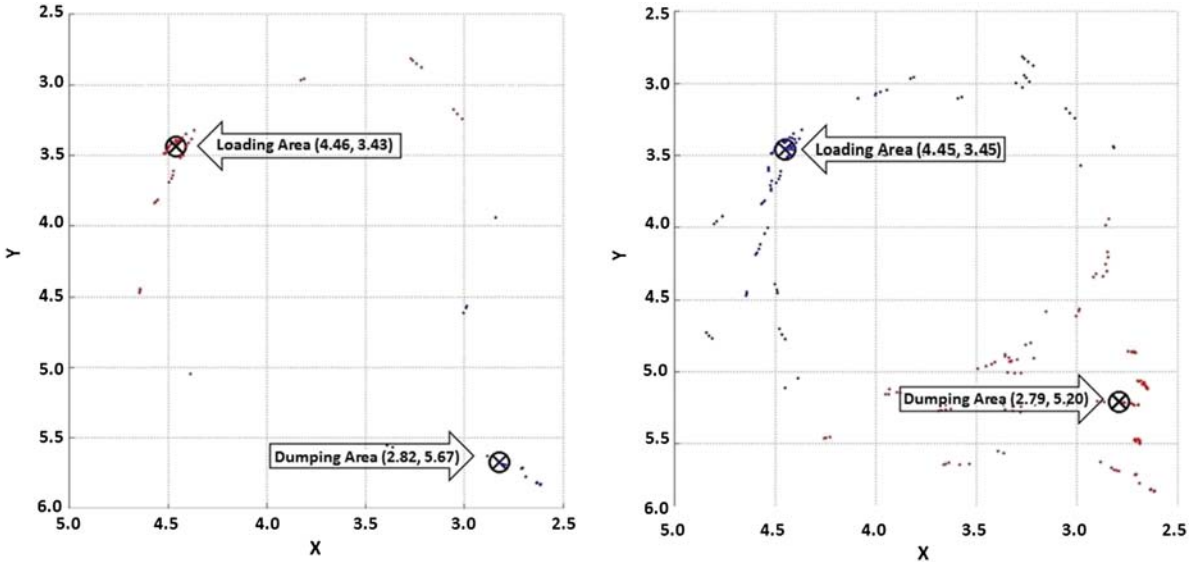

Fig. 5. Detecting changes in the location of cluster centroids over time

J. Constr. Eng. Manage.

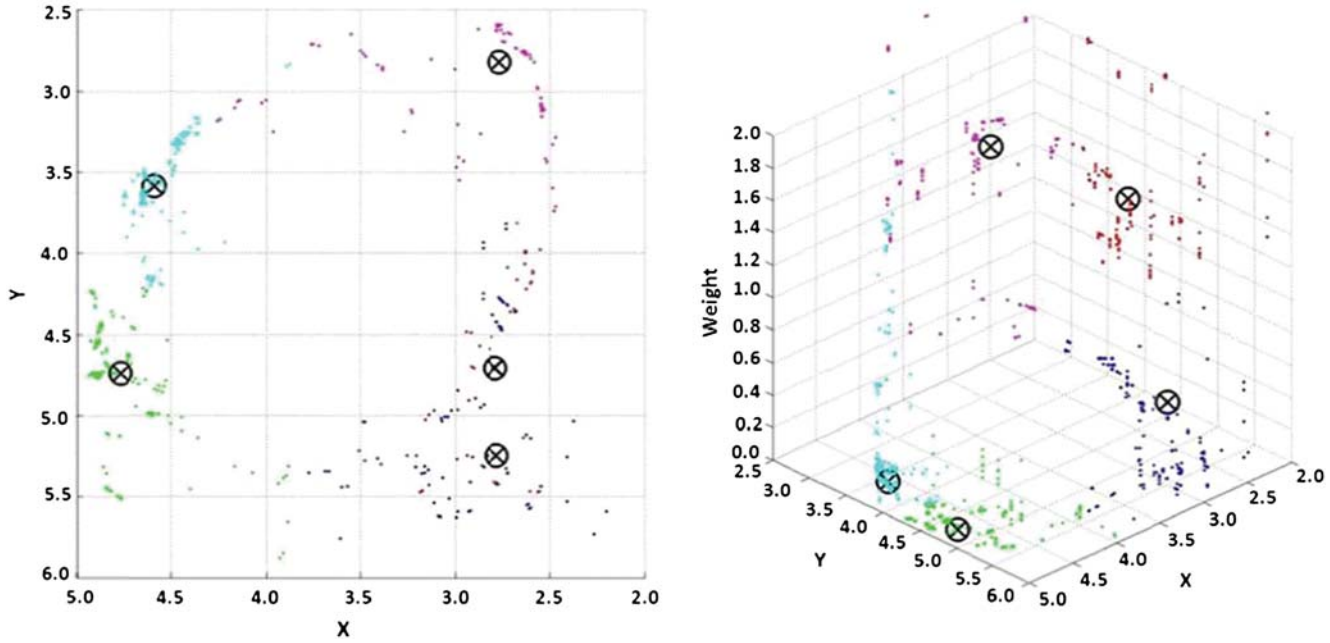

Fig. 6. Results of *k*-means clustering algorithm applied to the positional and weight data in 2D (*xy*) and 3D (*xyw*) spaces for Experiment 2

Experiment 2: One Loader and Multiple Dump Trucks, with Service Area

The second experiment consisted of multiple dump trucks, and thus it was expected that queues would form in the vicinity of loading and dumping areas. In particular, one front-end loader was tasked with loading three (two big and one small) dump trucks, one at a time. A designated service area was also added to test if the developed methodology can properly identify queues from this service area with no prior location information. Hence, it was expected that five clusters representing loading area, dumping area, loading queue, dumping queue, and service area were detected from the streaming fleet data. Therefore, the *k*-means algorithm was applied assuming $k = 5$. Fig. 6 illustrates the plots of collected positional and weight data in 2D (*xy*) and 3D (*xyw*) spaces. The developed *k*-means algorithm successfully detected five clusters and found the centroids as shown in Table 4.

Considering the load data trend, it was found that clusters 2 and 3 corresponded to the loading and dumping areas, respectively. Using the state taxonomy shown in Fig. 2, equipment states were then identified. Next, activity durations were calculated using time-stamped positional and weight data. Statistical analysis on pools of activity durations provided the mean and standard deviation of each activity duration. Table 5 compares observed values (from experiment video) with extracted values (using the reasoning process) of activity durations. Using Student's t-test, the t-statistic for load, haul, dump, and return activities are 0.69, 1.10, 1.16, and 1.69, respectively, and the *p*-values of all of them are greater than 0.05 (i.e., confidence level of 95%). Similar to Experiment 1, it can be concluded that there is no statistically significant difference between the two means.

Time-stamped data contained within the three clusters representing loading and dumping queues as well as the service area were also used to find the mean and standard deviation of resource waiting times in these locations. Table 6 compares observed values (from experiment video) with extracted values (calculated using the reasoning process and statistical analysis) of waiting times. Again, using the Student's t-test, the t-statistics of 1.89, 1.78, and 1.56 are calculated for the waiting times inside the loading queue, dumping queue, and service area. These t-statistic values account for *p*-values greater than 0.05 (i.e., confidence level of 95%).

Table 4. Centroids of the Detected Clusters Using *k*-Means for Experiment 2

Identified cluster	*x* (m)	*y* (m)	*w* (g)
Cluster 1	2.754550	4.42422	1.460313
Cluster 2	4.608870	3.53820	0.000792
Cluster 3	2.929420	5.36794	0.030927
Cluster 4	2.764165	2.77719	1.446068
Cluster 5	4.773650	4.73571	0.001201

Table 5. Observed versus Extracted Activity Duration Means and Standard Deviations for Experiment 2

	Observed duration (s)		Extracted duration (s)	
Activity	Mean	Standard deviation	Mean	Standard deviation
Load	22.3	10.5	25.7	11.5
Haul	30.9	8.90	34.7	6.33
Dump	10.1	4.80	8.10	2.53
Return	28.2	3.60	35.0	12.2

Table 6. Observed versus Extracted Means and Standard Deviations of Waiting Times for Experiment 2

	Observed waiting time (s)		Extracted waiting time (s)	
Location	Mean	Standard deviation	Mean	Standard deviation
Loading queue	51.5	15.5	39.4	12.9
Dumping queue	7.10	1.20	8.30	1.76
Service area	77.7	4.50	81.0	4.91

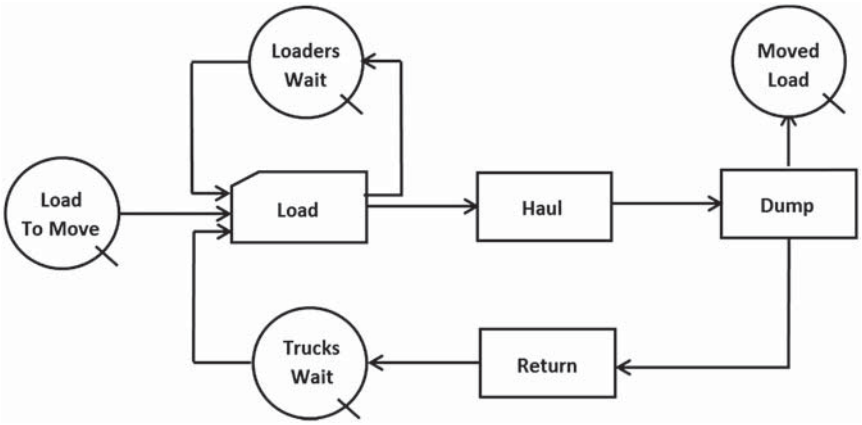

Fig. 7. ACD of the DES model of a typical earthmoving operation

Table 7. Approximated Activity Durations based on Overall Site Layout and Resource Specifications

Activity	Approximated duration (s)
Load	10.0
Haul	25.0
Dump	5.00
Return	20.0

Data-Driven Simulation

As stated previously, in order to make a transition from human-centered decision making to simulation-centered decision making, a sufficient amount of project information must be incorporated into simulation modeling. Hence, the next step in validating the results obtained from this research was to evaluate if the outcome can be used to generate more accurate simulation output. This step was essential because a robust and reliable data-driven modeling strategy that can safely replace human assumptions when simulating an engineering system is the first step toward enabling automated generation of DES models. Therefore, results obtained from the first experiment were used to update activity durations inside a DES model that was created in state and resource based simulation of construction processes (STROBOSCOPE) (Martinez 1996). The ACD of an earthmoving operation is illustrated in Fig. 7. Two DES input scripts were generated from this ACD. The first script contained activity durations (Table 7) calculated using the actual site layout and resource arrangement (Fig. 3), distances between work areas, and resource travel speeds. The second script was created using activity durations extracted from construction fleet *xyw* data and without making any assumptions about the site layout and resource arrangement.

Also, in both DES scripts, it was assumed that the hourly costs of a dump truck and a loader were \$50 and \$135, respectively.

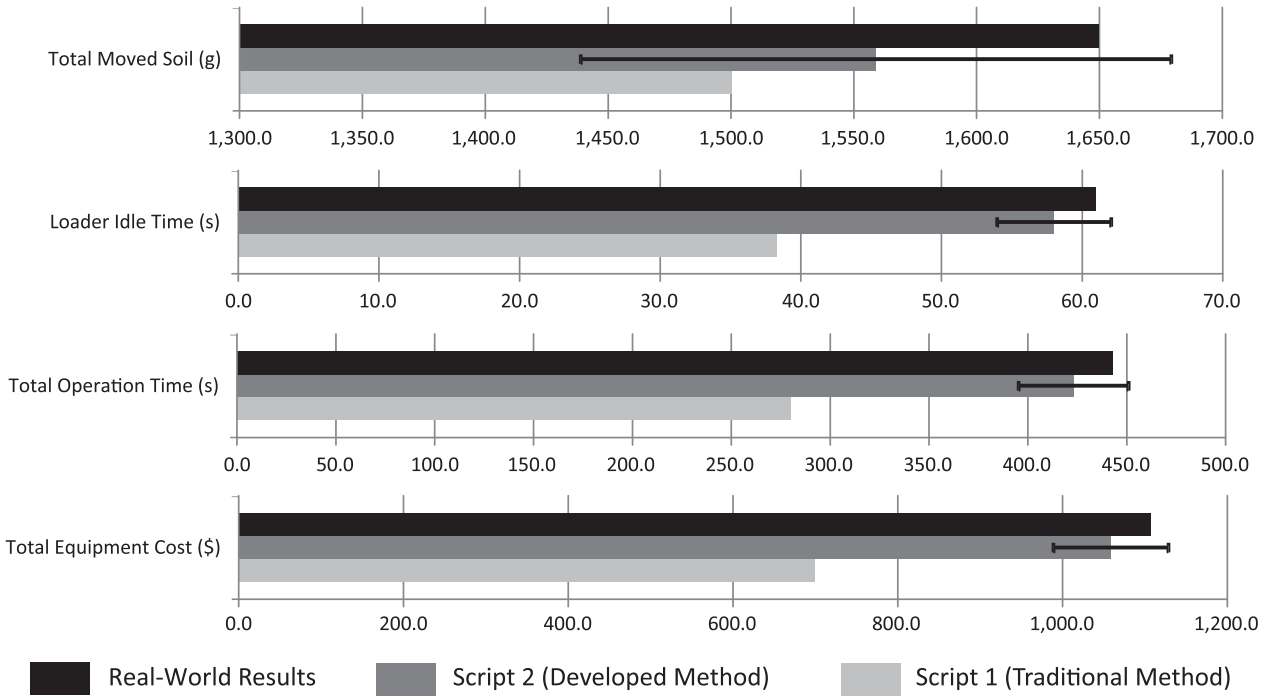

Fig. 8. Analysis of results obtained from the real-world experiment and the output of traditional and data-driven simulation models

As presented in Fig. 8, four separate measurable quantities (namely, total amount of transported soil, loader idle time, total operation time, and total equipment cost) were selected to assess the precision of results generated by the two simulation scripts.

The error bars for the values obtained from the simulation model based on the second script are also shown in this figure to indicate the statistical significance of results. As illustrated in Fig. 8, the output of the simulation model based on the second script (created using extracted duration values) with regard to all four measures was in closer agreement with the observed values from the real experiment. For instance, the total amount of transported soil in the real experiment was observed to be 1,650.0 units (in this paper, grams) in five cycles. When this operation was simulated, the output of the first script (created using engineering judgments and resource arrangement) indicated this quantity to be 1,500.5 units, while the output of the second script (created using the developed data collection and mining methods) indicated the same quantity to be 1,559.1 units with a standard deviation of 120.2 units. The same trend was also observed in the other three measures in which the output of the second simulation script was statistically very close or identical to that of the real operation.

Conclusions and Future Work

The convenience of making project decisions based on human expert judgments has caused the construction industry to remain to a large extent reluctant to the prospect of replacing human-centered with simulation-centered decision making. A contributing factor to this problem is that very often simulation models are made when little information is known about a project, rarely updated as the project makes progress, and thus not considered reliable and credible decision-making tools. To overcome these challenges, a systematic approach is needed to enable such models to continuously communicate with the real system, learn from the dynamics of events as they evolve, and accordingly adapt themselves to these changes. To this end, the authors investigated the prospect of enabling knowledge-based data-driven simulation model generation and refinement for construction operations. The main contribution of this research to the body of knowledge is that it lays the foundation to systematically investigate whether it is possible to robustly discover computer-interpretable knowledge patterns from heterogeneous field data in order to create or refine realistic simulation models from complex, unstructured, and evolving operations such as heavy construction and infrastructure projects. The aim of this paper was to report on the latest findings of this research by presenting a multimodal (i.e., position, weight, angle) process data mining, fusion, and reasoning algorithm capable of extracting operational knowledge and automatically updating the corresponding DES model. A statistical data point clustering algorithm based on the k-means method was also employed in conjunction with data-mining techniques to discover knowledge about the construction site layout and arrangement of resources. In order to validate the developed methodology, several experiments were conducted. Results indicated that extracted knowledge (e.g., activity durations, resource interactions, site layout) were valid and in good agreement with the reality of the project. Moreover, an earthmoving scenario was modeled in STROBOSCOPE and refined using the discovered operational knowledge, and a comparative analysis was conducted. The analysis revealed that results obtained from the DES script generated using extracted knowledge were more accurate and realistic compared to those from the DES script generated using human-made assumptions.

Future work will include the design and implementation of self-adaptive simulation models and a generic methodology for automated generation of data-driven simulation models capable of handling a wider variety of construction operational scenarios. The authors are currently working on the core components of a framework that integrates multimodal data collection and fusion and robust data mining to enable real-time manipulation of simulation models with operational knowledge extracted from incoming field data streams. The outcome of this work is sought to be validated in large-scale settings. The findings will be ultimately made available to the academic community and industry practitioners in an effort to disseminate the use of simulation models that realistically represent real world operations and their attributes.

References

AbouRizk, S., and Shi, J. (1994). "Automated construction-simulation optimization." *J. Constr. Eng. Manage.*, 120(2), 374–385.

Akhavian, R., and Behzadan, A. H. (2011). "Dynamic data driven simulation of ongoing construction operations." *Proc., 3rd Int. and 9th Construction Specialty Conf.*, CSCE, Ottawa, ON, Canada.

Akhavian, R., and Behzadan, A. H. (2012). "An integrated data collection and analysis framework for remote monitoring and planning of construction operations." *Adv. Eng. Inf.*, 26(4), 749–761.

Akinci, B., Boukamp, F., Gordon, C., Huber, D., Lyons, C., and Park, K. (2006). "A formalism for utilization of sensor systems and integrated project models for active construction quality control." *Autom. Constr.*, 15(2), 124–138.

Ali, R., Ghani, U., and Saeed, A. (1998). "Data clustering and its applications." ⟨http://members.tripod.com/asim_saeed/paper.htm⟩ (Jul. 23, 2013).

Andoh, A. R., Xing, S., and Hubo, C. (2012). "A framework of RFID and GPS for tracking construction site dynamics." *Proc., Construction Research Congress 2012, Construction Challenges in a Flat World*, ASCE, Reston, VA, 818–827.

Banks, J. (1998). *Handbook of simulation: Principles, methodology, advances, applications, and practice*, Wiley, New York.

Behzadan, A. H., Aziz, Z., Anumba, C. J., and Kamat, V. R. (2008). "Ubiquitous location tracking for context-specific information delivery on construction sites." *Autom. Constr.*, 17(6), 737–748.

Berkhin, P. (2002). "A survey of clustering data mining techniques." *Technical Rep.*, Accrue Software, San Jose, CA, 1–56.

Bishop, C. M. (2006). *Pattern recognition and machine learning*, Springer, New York.

Brilakis, I., Park, M. W., and Jog, G. (2011). "Automated vision tracking of project related entities." *Adv. Eng. Inf.*, 25(4), 713–724.

Caldas, C. H., Grau, D. T., and Haas, C. T. (2006). "Using global positioning system to improve materials-locating processes on industrial projects." *J. Constr. Eng. Manage.*, 132(7), 741–749.

Chen, P., Buchheit, R. B., Garrett, J. H. Jr., and McNeil, S. (2005). "Web-vacuum: Web-based environment for automated assessment of civil infrastructure data." *J. Comput. Civ. Eng.*, 19(2), 137–147.

Chung, T. H., Mohamed, Y., and AbouRizk, S. M. (2006). "Bayesian updating application into simulation in the North Edmonton Sanitary Trunk tunnel project." *J. Constr. Eng. Manage.*, 132(8), 882–894.

Daneshgari, P., and Moore, H. (2009). "The secret to short-interval scheduling." *Technical Rep.*, Electrical Construction and Maintenance, Chicago, IL, 32–36.

Davis, W. J. (1998). "On-line simulation: Need and evolving research requirements." *Handbook of simulation*, Wiley, New York, 465–516.

Duda, R. O., Hart, P. E., and Stork, D. G. (1973). *Pattern classification and scene analysis*, 2nd Ed., Wiley, New York.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). *Advances in knowledge discovery and data mining*, AAAI/MIT Press, Menlo Park, CA.

Ford, D. R., and Schroer, B. J. (1987). "An expert manufacturing simulation system." *Simulation*, 48(5), 193–200.

Gong, J., and Caldas, C. H. (2010). "Computer vision-based video interpretation model for automated productivity analysis of construction operations." *J. Comput. Civ. Eng.*, 24(3), 252–263.

Grau, D. T., and Caldas, C. H. (2009). "Methodology for automating the identification and localization of construction components on industrial projects." *J. Comput. Civ. Eng.*, 23(1), 3–13.

Hajjar, D., and AbouRizk, S. (1999). "Simphony: An environment for building special purpose construction simulation tools." *Proc, of the 31st Conf. on Winter Simulation: Simulation—A bridge to the future*, Vol. 2, Association for Computing Machinery (ACM), New York, 998–1006.

Han, S., Lee, S., and Peña-Mora, F. (2011). "Application of dimension reduction techniques for motion recognition: Construction worker behavior monitoring." *Proc., 2011 ASCE Int. Workshop on Computing in Civil Engineering*, ASCE, Reston, VA, 19–22.

Heidorn, G. E. (1974). "English as a very high level language for simulation programming." *Proc., ACM SIGPLAN Notices*, Association for Computing Machinery (ACM), New York, 91–100.

Jang, W. S., and Skibniewski, M. J. (2007). "Wireless sensor technologies for automated tracking and monitoring of construction materials utilizing Zigbee networks." *ASCE Construction Research Congress*, ASCE, Reston, VA.

Kannan, G., and Vorster, M. (2000). "Development of an experience database for truck loading operations." *J. Constr. Eng. Manage.*, 126(3), 201–209.

Lee, W. J., Song, J. H., Kwon, S. W., Chin, S., Choi, C., and Kim, Y. S. (2008). "A gate sensor for construction logistics." *Proc., 25th Int. Symp. on Automation and Robotics in Construction*, Institute of Internet and Intelligent Technologies, Vilnius, Lithunia, 100–105.

Ling, Q. (2011). "How many low-precision sensors are enough for reliable detection?" *IEEE Trans. Aerosp. Electron. Syst.*, 47(4), 3001–3006.

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

MacKay, D. J. C. (1992). "Information-based objective functions for active data selection." *Neural Comput.*, 4(4), 590–604.

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

Martinez, J., and Ioannou, P. G. (1994). "General purpose simulation with stroboscope." *Proc., 1994 Winter Simulation Conf. (WSC)*, Association for Computing Machinery (ACM), New York, 1159–1166.

Martinez, J. C. (1996). "Stroboscope: State and resource based simulation of construction processes." Ph.D. dissertation, Univ. of Michigan, Ann Arbour, MI.

Mathewson, S. C. (1984). "The application of program generator software and its extensions to discrete event simulation modeling." *IIE Trans.*, 16(1), 3–18.

Mirkin, B. (2005). *Clustering for data mining: A data recovery approach*, Chapman and Hall/CRC, Boca Raton, FL.

Navon, R. (2005). "Automated project performance control of construction projects." *Autom. Constr.*, 14(4), 467–476.

Park, M. W., Palinginis, E., and Brilakis, I. (2012). "Detection of construction workers in video frames for automatic initialization of vision trackers." *Proc., Construction Research Congress 2012, Construction Challenges in a Flat World*, ASCE, Reston, VA, 940–949.

Pradhan, A., and Akinci, B. (2012). "A taxonomy of reasoning mechanisms and data synchronization framework for road excavation productivity monitoring." *Adv. Eng. Inf.*, 26(3), 563–573.

Razavi, S. N., and Haas, C. T. (2012). "Reliability-based hybrid data fusion method for adaptive location estimation in construction." *J. Comput. Civ. Eng.*, 26(1), 1–10.

Rezazadeh Azar, E., and McCabe, B. (2012). "Vision-based recognition of dirt loading cycles in construction sites." *Construction Research Congress*, ASCE, Reston, VA, 1042–1051.

Sall, J., Lehman, A., Stephens, M., and Creighton, L. (2012). *JMP start statistics: A guide to statistics and data analysis using JMP*, SAS Institute, Cary, NC.

Soibelman, L., and Kim, H. (2002). "Data preparation process for construction knowledge generation through knowledge discovery in databases." *J. Comput. Civ. Eng.*, 16(1), 39–48.

Son, Y. J., and Wysk, R. A. (2001). "Automatic simulation model generation for simulation-based, real-time shop floor control." *Comput. Ind.*, 45(3), 291–308.

Song, L., Cooper, C., and Lee, S. (2009). "Real-time simulation for look-ahead scheduling of heavy construction projects." *Proc., Construction Research Congress*, ASCE, Reston, VA, 1318–1327.

Véjar, A., and Charpentier, P. (2012). "Generation of an adaptive simulation driven by product trajectories." *J. Intell. Manu.*, 23(6), 2667–2679.

Yang, J., Arif, O., Vela, P., Teizer, J., and Shi, Z. (2010). "Tracking multiple workers on construction sites using video cameras." *Adv. Eng. Inf.*, 24(4), 428–434.

Yuan, Y., Dogan, C. A., and Viegelahn, G. L. (1993). "A flexible simulation model generator." *Comput. Ind. Eng.*, 24(2), 165–175.