ARTICLE IN PRESS
AUTCON-02070; No of Pages 14

Automation in Construction xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changing site layout

Amr ElNimr, Muaz Fagiar, Yasser Mohamed *

Hole School of Construction Engineering, Department of Civil and Environmental Engineering, University of Alberta, 3-011 Markin/CNRL Natural Resources Engineering Facility, Edmonton, Alberta, T6G 2W2, Canada

ARTICLE INFO

Article history:
Received 14 May 2015
Received in revised form 22 February 2016
Accepted 7 May 2016
Available online xxxx

Keywords:
Simulation
3D visualization
Distributed simulation
Pathfinding
Site layout

ABSTRACT

Construction site layout changes dynamically depending on the progress of construction activities in ways that may constrain access routes and paths of mobile resources. Discrete event simulation is a useful tool for modeling time dependent activities and resource interactions but usually cannot depict spatial changes in a natural way. This paper describes a framework for integrating 3D visualization components with discrete event simulations to model the movement of mobile cranes on industrial construction projects and facilitate tempo-spatial planning of site layout. The paper focuses on the integration approach using distributed simulation standards and spatial analysis using a dedicated visualization component with pathfinding algorithm. The A* algorithm is used in conjunction with mesh generation mechanism for developing the proposed framework. The benefits of the proposed framework emerge from taking full advantage of the geometrical and special processing strengths of the visualization component through loose coupling with simulation components. The paper describes the main components of the framework and the two-way communication mechanism between them based on distributed simulation standards. It also presents a case example of a real world project to demonstrate the proposed framework and compare the outcome to a traditional solution that uses a probability distribution to model crane travel times.

1. Introduction

Construction activities affect the space of construction sites over time. Therefore, models of such activities should account for the dynamic interaction between resources and space over time. The use of discrete event simulation is effective in representing construction process logic, resource interactions and uncertainty in activity durations [1,2] but usually unable to represent construction site space utilization in an intuitive way. By comparison, 3D visualization technologies have the ability to represent space and site layouts in a natural and intuitive way, understandable by various project parties. The integration of simulation and 3D visualization in a single framework has its benefits as indicated through several publications [3–6]. However, challenges and limitations exist in previous integration approaches, which constrain full utilization of 3D modeling and visualization technologies [7,8]. In particular, feeding back results of geometry and space analysis of construction sites into simulation models at different time intervals of project execution (i.e. active participation of visualization components in simulation) is very limited. Integrating these two modeling techniques (i.e. simulation and 3D) without compromising their strengths can provide a valuable tool for dynamic planning of site space and resource mobilization throughout a project's lifecycle.

The work described in this paper uses a distributed simulation framework to integrate time-related site space (tempo-spatial) changes with simulation modeling. The framework is utilized here to model resource mobilization behaviors related to a resource's need of a certain space at a certain scheduled project time in order to perform its scheduled event. The case of heavy lift mobile cranes on industrial construction site is used to demonstrate the structure and utility of the framework. In this case the tempo-spatial behavior of these expensive resources include the need to move from one lifting location to another during the progress of the project, which in turn requires finding the shortest, obstacle-free paths under a dynamically changing site, where additional parts of the final facility are added at different time intervals.

This paper discusses the background of dynamic site space modeling and Simulation Driven Visualization (SDV) mechanisms, going through the relevant construction research related to both topics. It also discusses search algorithms, with emphasis on the A* algorithm and its use for path finding. The paper then presents the implementation of

---

* Corresponding author. Tel.: +1 780 492 1321; fax: +1 780 492 0249.
E-mail address: yaly@ualberta.ca (Y. Mohamed).

http://dx.doi.org/10.1016/j.autcon.2016.05.013
0926-5805/© 2016 Published by Elsevier B.V.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

the proposed framework in the case of heavy lift mobile cranes in a real construction project.

1.1. Main contributions

The high cost and long time for mobilization processes of heavy resources, such as cranes, have been the subjects of many construction related studies. A stand-alone simulation system does not have the ability to neither represent site spatial changes in an intuitive way nor plan ahead for various resources' routes based on tempo-spatial interactions of those resources and the dynamically changing site layout. Accelerating this process by ascertaining the shortest mobilization path and accurately estimating its duration, considering dynamically changing site layouts, are invaluable.

Equally important, the current state of the art in simulation driven visualization models has a number of characteristics that hinder its utilization for day-to-day decision-making. It does not meet the growing complexity of construction operations and the need to integrate heterogeneous simulation, computing algorithms, and data sources in these models. These characteristics include: 1) One way data flow as existing SDV mechanisms do not allow the simulation to receive back information from the visualization; limiting the user/decision maker from evaluating or adjusting various construction processes and site layout scenarios based on the visualization output. 2) Coupling between simulation and visualization engines. This leads to compromises on the strengths of both the simulation and visualization components and decreases their reusability in day-to-day construction operations. Running the two components in parallel increases the demand on computer hardware and limits the development of highly detailed simulation and visualization models. In addition, it hinders the utilization of specially developed software for handling graphics and visualization tasks. 3) Post-processing visualization as it does not allow the user/decision-maker to interact with the simulation.

The proposed framework is based on the High Level Architecture (HLA) standards for distributed simulation with a Pathfinding Mechanism Extension (PME) to model site spaces geometry and find the shortest safe routes based on changes in these spaces. The main challenges in applying the proposed framework and the solutions developed to overcome these challenges focus mainly on: 1) establishing two-way communication and synchronization between the simulation and visualization components, enabling space and path analysis inside the visualization component, and 2) decoupling the visualization component and the simulation to take full advantage of their strength independently and exclusively.

The work thus promotes the adoption of distributed simulation by wider construction researchers, demonstrating how open communication along with separate and independent execution of the components can enable simulation developers to embed variety of algorithms within visualization components taking full advantage of their 3D object models and processing engines.

2. Research background

Researchers have used various methods to model changes in site space with time (dynamic site layout planning). The following sections summarize those research efforts to demonstrate the novelty of the approach followed in this study.

2.1. Site space modeling

Site layout planning is an important task that involves the positioning of temporary facilities and structures onsite to minimize their associated costs and correspondingly the overall project costs [3]. Site layout planning can be categorized into static and dynamic categories. Static site layout planning models produce a single site layout that identifies static locations for a project's temporary facilities. Those facilities are fixed throughout the project's execution; accordingly, this type of layout planning is not influenced by site space changes and availability [9]. Simulation has been one of the methodologies adopted for static layout modeling [10,11].

Dynamic layout planning is a similar technique to what the present methodology proposes as it involves the changes in a site's available space resulted from either building new temporary facilities or relocating existing ones [9,12]. The models that deal with dynamic site layout planning in construction are limited [13]; among the most common of these are genetic algorithms [13], 4D modeling [14,15], dynamic programming [9] and Computer Aided Design [16].

4D-based dynamic site layout models inherit the limitation associated with 4D modeling that is depicting changes in site space only on the basis of overall construction product progress. Rather, 4D-based models consider space changes inside the constructed product itself; such as the layout inside each constructed floor [15]. Unlike the proposed SDV-based model, they do not consider the effects of site space changes on mobile resource movement throughout the project execution. Both genetic algorithm (GA) and 4D-based models have limitations in depicting the changes in site space that occur as the building structure (fixed facility) evolves. For example, the footing excavation of a certain group of buildings starts at a later date in a residential complex project, so the site space for future excavations can be utilized during the early stages of the project.

Researchers have attempted to improve the use of these dynamic site layout tools. Tommelein et al. [17] introduced MovePlan that is a dynamic layout tool with graphical user interface. MovePlan generates optimized site layouts by using the activity relationships as inputs. A GIS-based construction site material layout evaluation tool was also developed by Su et al. [18] which calculates the material accessibility grade on a construction site taking the resource loaded schedule as input.

Those models provide snapshots of different optimal site layouts for each stage of the project by adopting a chronological procedure [9]. However, in reference to models utilizing GA and 4D modeling, El-Rayes and Said [9] stated that the layout's efficiency; at the later project stages, is greatly affected by decisions taken in early layouts, which does not guarantee a global optimal solution. Also, these approaches may provide infeasible solutions as early-located facilities may cause insufficient space for future facilities [9].

In their research El-Rayes and Said [9] addressed the “Local Optimal site layout problem” by using dynamic programming epochs (rules) to produce an overall optimal site layout plan that takes into consideration changes in temporary facilities at future project dates. Their model relies on the Manhattan distance to calculate movement rate; however, this approach is not necessarily the shortest, safest and cost effective traveling path. The authors suggested that their model can be expanded in future research to support 3D modeling of site layouts, detailed planning of resource travel paths, and considering the dynamic impact of material procurement decisions on site storage and layout planning.

Said and El-Rayes [19] proposed a congested construction logistic planning (C2LP) model which generates site layout plans and optimal material logistics. Parameters such as the site exterior and interior spatial data, temporal facilities' dimensions are required by C2PL to optimize the storage location in exterior and interior building spaces. Researchers also started using BIM models to facilitate site layout planning as they are rich source of information [20,21]. In another research, Said and El-Rayes [22] attempted to automate the retrieval of spatial and temporal data from Building Information Models (BIMs) in their developed construction logistic optimization system. Kumar and Cheng [23] further improved the practicality of the current tools by proposing a BIM based framework that automates the creation of mathematical models for dynamic site layout planning. Their framework utilize information from BIM in congestion with A^* algorithm and genetic algorithms; allowing its users to update design and schedule changes at a click of a button.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

Sadeghpour et al. [24] investigated the “snapshot issue” by providing a set of consecutive layouts throughout the project. Although their research showed a creative way of solving this problem, it lacked the concurrent visualization component to depict both: the changes in site space and the dynamicity of these layout changes. In other recent research, site space modeling was used to address the various site spatial needs for resources; mainly equipment [25,26]. This research focused on classifying resources and setting their choice criteria based on their space needs in comparison to the modeled site spatial data. Another more relevant research was done by Albahnassi et al. [27] that relate space geometry to heavy mobile equipment motion (mobilization) planning. This research; however, concentrated more on finding safe routes for equipment.

2.2. Simulation driven visualization in construction research

Recognizing the 4D modeling shortfall of not depicting resource interactions during the construction, researchers started to integrate construction simulation models with 3D Computer-aided Design (CAD) for animation to achieve a post-process playback of the simulation [1,2,28]. Good examples are the “Utopian Framework” for earth moving operations and the introduction of product-based Simulation Environment that build simulation models using product models based on CAD [29,30].

Akbas [4] introduced the Geometry-based Process Model (GPM) approach that uses discrete-event systems and simulation to create 3D elements of constructed products with geometric transformations. Kamat and Martinez [3,5] used the trace file feature in simulation models to invoke post-simulation animation of the simulation. They introduced the Dynamic Construction Visualizer (DCV) and Visualization of Simulated Construction Operations (VITASCOPE). Other researchers used this approach for design reviews, conflict detection and construction scheduling of tunneling projects [31].

It is also proven to be useful for earthmoving operations. Researchers used this approach to develop earthmoving simulators proving training for a variety of activities such as excavation, trucks' loading and unloading processes and heavy duty machine control [32,33].

There are also some commercial packages that exist in the market for simulation visualization, such as Delmia's Quest® and Brooks Software AutoMod®; however, they are generally concentrated on manufacturing operations. These packages fall short when handling complex construction operations as they will require an essential change in the model conceptualization from construction model developers [5,36]. A package used in the construction industry is CATIA® that combines visualization and CAD packages. It is powerful and effective in handling CAD aspects; however, it lacks the DES which makes it a better solution for 4D modeling.

On the other hand, Discrete Event System Specification (DEVS) formalism was introduced; providing a framework for defining hierarchical discrete-event models in a modular way. However, the DEVS-based toolkits developed require extensive knowledge and expertise of advanced programming techniques. By taking advantage of this framework as well as overcoming the aforementioned challenge, Wainer and Liu [37] introduced a state-based graphical modeling paradigm that is based on CD++Modeler toolkit to analyze simulation data using 2D graphics. Also, they presented other 3D animation tools such as CD++/Maya and CD++/Blender. These toolkits allow its users to navigate and interact with animated 3D models to investigate simulation data in situations such as emergency response planning and serious gaming. This methodology has gained increasing popularity as by decoupling the model and simulation concept, the same model can be executed on different simulators, which can be independently verified and reused [38].

As technology advances, researchers also started to look into distributed simulation to solve the interoperability. High Level Architecture (HLA) standards provide a framework for simulation interactions regardless of their computing platforms. In this approach, researchers usually develop and allocate a simulation component to merely visualize the simulated construction behavior. AbouRizk [6] used this approach to visualize tunneling construction. Other researchers also used it for simulating earthmoving operations to visualize trucks' graphical movement and display statistics on some of their performance indicators [39]. Elnimr and Mohamed [8] exploited the distributed simulation approach to integrate game engines to visualize pipeline operations. Also, ElNimr and Mohamed [40] presented a loosely coupled visualization of industrial construction operations that visualizes the construction process logic and presents changes in a site's spatial aspects. Moreover, recent efforts by Rekapalli and Martinez [41,42] have focused on achieving two-way communication and user interaction with the visualization component of a discrete event simulation in a non-distributed way. Behzadan et al. [43] conduced a vigorous review on DEVS and HLA systems highlighting their strength and underlying challenges.

2.3. Data driven simulation models

Recent studies have addressed the longstanding challenge of generating adaptive simulation models that are responsive to the project dynamic changes during the construction stage. Recent efforts took advantage of tracking technologies to capture trucks and excavator motions and enhance the detection of equipment state [34,35]. Song and Eldin [44] highlighted the ineffectiveness of using statistical input data for simulation models to analyze look-ahead schedules, which, for precision, require the most recent project performance data on a real-time or near-real-time basis. The authors proposed an adaptive real-time tracking and simulation of heavy construction operations using sensors to constantly capture and feed the dynamic site condition changes into the simulation for more accurate look-ahead scheduling. Akhavian and Behzadan [45,46] employed data mining methods to extract contextual knowledge from heterogeneous field data that are captured through ubiquitous sensors to automatically generate and refine a simulation model. Further, the authors used built-in smartphone sensors to detect data other than positional information to recognize construction equipment activities [47]. Although, all such studies provided means of realistic input data for simulation models, their utilization for simulation driven visualization is limited with one-way data flow; hindering the full benefit of real-time visualization models.

2.4. The need for resources' site tempo-spatial planning framework

Heavy resource mobilization in a dynamically changing site layout can be a costly and time-consuming process involving many variables. For example, the planning of crane lifts of various components of an industrial facility involves several mobilizations of multiple heavy-lift cranes (over 400 tons). Mobilizing these cranes has to consider the dynamic changes in construction site space. It is a difficult task to ascertain the shortest and obstacle-free mobilization route for each of these cranes in a changing dynamic site layout as well as estimate the mobilization duration required for each of the module's lifting activity.

Construction process simulation alone does not have the ability to represent site spatial changes in an intuitive way. A typical solution is representing space as a resource inside the simulation model, yet this can be counter-intuitive and difficult to visualize by a project team. For example, the simulation model would depict a crane only as a resource with a certain quantity without showing the crane maneuverability space or the crane relocating routes. Also, a stand-alone simulation system does not plan ahead for various resources' routes based on tempo-spatial interactions of those resources and the dynamically changing site layout. A planning method is needed to predict these changes in site layout and to choose and evaluate the various mobilization routes for each of the cranes.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

Recognizing these needs, ElNimr and Mohamed [7,8], developed a framework (will refer to it as HSV in short for HLA based simulation-driven visualization) that depicts modules' construction and check for any flaws in their installation sequence. The framework uses a visualization component (Blender game engine (BGE)) [48] to depict a dynamically changing site layout based on inputs from a simulation component (HLA federate) [7,8]. The framework was also utilized to check the availability of modules' assembly bays on an off-site assembly yard. In both cases, the framework depicted the space availability in both the module assembly yard and on the construction site.

This paper further enhances the simulation site space representation in the framework by adding a two-way communication mechanism between the simulation and visualization components. This feature enabled the implementation of a pathfinding mechanism inside the visualization component (BGE) to (a) assess feasible routes for each heavy-lift mobile resource mobilization, (b) find and visualize the shortest (obstacle-free) path for each mobilization event and (c) calculate the expected duration of the mobilization.

It is important to note that static layout planning is not the focus of this research; the scope is to model changes in site space during the project execution using a SDV framework and not producing a snapshot in time of an optimal site layout. This proposed framework does not produce single or multi-optimal site layouts, but simulates and depicts the changes in site space with time through each stage of the project. This is done through integration between a discrete event (DES) engine and a 3D engine with embedded path finding and two-way communication capabilities. The HSV framework takes into consideration changes in site space due to both temporary facilities' and permanent structural elements' locations when depicting layouts and calculating the shortest resource routes throughout the project's lifecycle.

3. Pathfinding for mobile resource mobilization

3.1. Formulating mobile resource pathfinding problem as a search problem

Search in artificial intelligence refers to an object (agent) examining different possible sequences of actions that lead to states of known values, then choosing the best sequence based on the desired search criteria. The problem is to dynamically model changes in site space throughout the project execution to achieve the goal of finding the shortest obstacle-free path when moving a mobile resource from one location to another on the current site layout. A search algorithm takes this problem as input and returns a solution in the form of an action sequence. As per Russell and Norvig [49] approach of defining problems and solutions, the problem can be formally defined by the following components:

Initial State: The mobile resource's initial location at the current site layout, which can be expressed as "In<LocationID>". For example, if a mobile crane is at lift Location 5, then its initial state is "In<5>".

Goal State: The mobile resource's goal location. For example, if a crane needs to be moved from Location 5 to lifting Location 9, then its Goal State is "In<9>".

State Space: The set of all states reachable by a mobile resource from the initial state. For example, if a crane is at a certain site location represented by certain coordinates, the State Space is all the other locations (states) that can be reached from this location. The State Space can be connected through a graph, which can be represented by either a tree or grid. In our case the grid was chosen to cover the site layout due to the reasons discussed thoroughly in Elnimr [18].

Path Cost: A function; explained in the next section, that assigns cost to each resource path. The Path Cost function gives a cost as the pathfinding mechanism tries to obtain the shortest path for moving mobile resources.

Actions: The set of actions that a resource can take in order to move from one state to another before ultimately reaching the Goal State; if a solution was found.

Goal Test: Tests to determine whether the resource state is its Goal State.

3.2. A* search algorithm

There are a number of search algorithms used for path planning such as Probability Roadmaps (PRMs) [50] and Randomized Path Planner [51] that are used for robot path planning. The choice of a particular search algorithm in this work is not a primary objective. Instead, a more primary goal is to demonstrate the ability of the framework to accommodate a sophisticated search algorithm that depends on geometrical attributes in a 3D scene in a way that is fully enclosed in the visualization component and independent of the DES component. If such separation is possible, as is the case in this work, replacement of the search algorithm with more advanced or efficient ones can be done without affecting other parts of the framework. Nevertheless, an informed search algorithm was preferred over an un-informed search for several reasons.

Un-informed search algorithms, also called blind search, cannot tell if any state of the searching object is more promising than another in terms of reaching the goal state (lifting location for resource). Also, Un-informed search techniques usually use an explicit search tree as their State Space while informed search algorithms (like A* algorithm) have the ability to use graphs as their search space. Lastly, Un-informed search strategies can be computationally expensive yet not reaching an optimum solution. Russell and Norvig [49] set out with 4 criteria to assess the various search algorithms: (1) Completeness; does the algorithm find a solution when there is one, (2) optimality; does it find the optimal solution (the shortest path in our case), (3) time complexity; how long does it take to find a solution in terms of processing time, and (4) space complexity; how much memory is needed to perform the search. The major types of un-informed search algorithms were evaluated based on these criteria. Un-informed algorithms usually have high time and space complexities that can hinder (slow down) the communication and time management between the simulation and visualization components of the framework. The un-informed algorithms that might be low on time and space complexity are not guaranteed to be complete or optimal in finding the shortest path. Informed search algorithms, on the other hand, are complete and optimal, provided that the heuristic part of the evaluation equation $h(\mathrm{N})$, does not overestimate the distance to the target.

A* (A star) is a widely used graph-search algorithm, developed in 1968 by Peter Hart, Nils Nillson, and Bertram Raphael [52]. It is an informed search algorithm; meaning the algorithm can differentiate if any of the states of a resource are better (more promising) than others in reaching the goal state. Also, it depends on an evaluation function $f(N)$ to decide on the mesh or graph node to expand to the next. This $f(N)$ function; illustrated in Eq. (1), measures the distance to the goal, and the lowest evaluation is selected for expansion.

$$f(N) = g(N) + h(N) \tag{1}$$

where,

N	node being processed by the algorithm
$f(N)$	A* evaluation function that represents estimated cost of the solution (shortest path) through node N.
$g(N)$	the cost to reach the current node N from start node.
$h(N)$	estimated cost of the shortest path from node N to the goal node N_G.

The $g(N)$ function measures the exact distance from the start node to current processed node on a graph. The $h(N)$ is a heuristic function and a

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

key component of this algorithm as it estimates the distance from the current node to the goal node on the site mesh. As it can be noticed, the A* algorithm takes into consideration the distance from the start node to the current node, in addition to the heuristic component of the equation. The way in which this algorithm solves the resources' pathfinding problem inside the HSV framework will be explained in Section 6. Moreover, some criteria were used in preference of A* algorithm rather than other pathfinding algorithms. First, It has an exact cost function $g(N)$ and heuristic function $h(N)$. Thus it is able to avoid the shortcomings of informed algorithms that depend only on heuristic functions which usually underestimate the path cost [17,49]. Secondly, A* takes into consideration the distance already traveled by the resource. This is ideal to resource traveling problems since an educated estimate of a resource's total movement is needed, not only a segment of the movement. Among all the algorithms that are considered optimal, there is no other algorithm that is guaranteed to expand to fewer nodes before reaching the shortest path than the A* algorithm [17,49]. This benefits the simulation run by avoiding slowdowns and data processing/flow bottlenecks. This provides fast processing of shortest paths, sending timely feedback on resource movement planning to the simulation component of the framework.

It is worth mentioning that the A* search algorithm will give the shortest paths between feasible nodes on a mesh, but it cannot detect if a node becomes infeasible because of obstacles resulting form site layout changes. To overcome this, another mechanism was implemented; the mesh generation mechanism, to work hand-in-hand with the A* search algorithm. This mechanism will change the generated node mesh every time the simulation changes the site layout; a detailed explanation is provided in Section 6.1.

4. Framework for resources' site tempo-spatial planning

During the construction, there are various objects with variant geometries occupying the site spaces. These objects can be; permanent structures (for example structural elements) or temporary site facilities occupying site space as the construction proceeds. All these objects cause changes in both; the site space geometry and mobile resources paths on site. Modeling a site's spatial data in an intuitive way inside simulation and then finding the heavy lift resources' shortest safe travel paths on a dynamically changing site layout with changing geometries is the problem this framework is trying to address.

Fig. 1 illustrates the framework that implements the proposed algorithms for resources' site tempo-spatial planning. The HSV framework is employed to model changes in site space with time, and study their effects on resource paths onsite using a pathfinding mechanism implemented inside the visualization component.

The proposed distributed Simulation Visualization framework is based on the High Level Architecture (HLA) standards [53]. HLA is a set of standards that regulates distributed simulation development in which the simulation is decomposed into smaller components that are linked together over computer networks to interact and synchronize (or interoperate) during a simulation run through Runtime Infrastructure (RTI) software [54]. The proposed framework uses Construction Synthetic Environment (COSYE) that is an application programming interface which supports the development of large-scale distributed synthetic simulation environments. The simulations or models connected to the COSYE are called Federates and a collection of them working together to simulate the same system is known as a Federation. A federate is not required to be a simulation model, but can be any piece of software or hardware, or even an interface to human user (user-in-the-loop), as long as each federate complies with the HLA standards. RTI facilitates communication between federates and provides HLA services such as simulation time management and object ownership management. A key component of a Federation is the Federation Object Model (FOM) [55] which defines all entities produced by the simulation, and documents the information that work in publish-subscribe model.

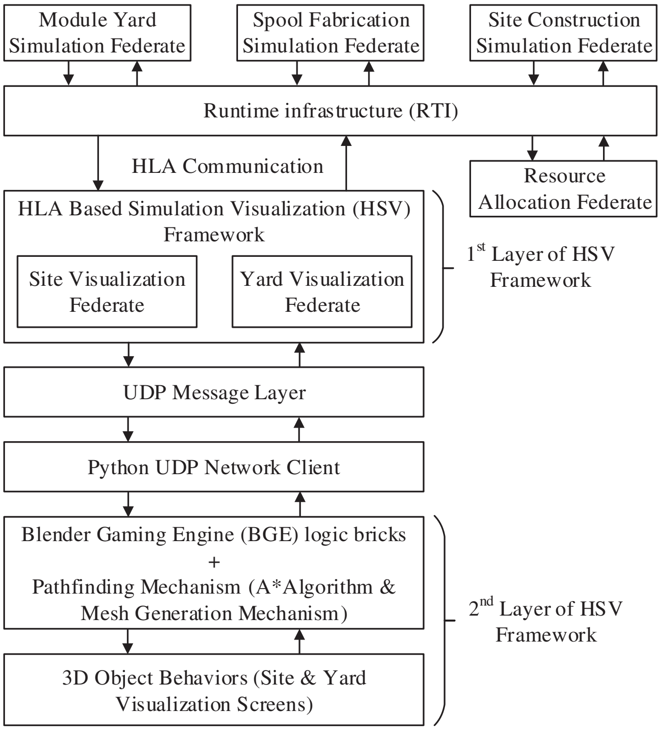


Fig. 1. Resource's site tempo-spatial planning framework.

As shown in Fig. 1, the following are typical components that are part of the framework or can be connected to it:

- Visualization Federate(s): It depicts the construction operation simulation concurrently as the simulation time advances to reflect the simulation behavior instantly. It subscribes to the simulation object (cranes and modules) attributes published by any of the other simulation federates and communicates their values with the secondary layer of the framework.
- Simulation Federate(s): That is where the simulation of the various construction processes takes place such as spool fabrications and spool assembly. Also, simulation time is usually advanced by these federates. They connect to their respective Viewer Federate through the RTI.
- Other Functional Federates: Any other federates that may provide supporting services to the federation. For example, the resource allocations federate that is responsible for allocating the various cranes to lift/move modules to their locations on site, based on the availability and fitness of cranes to handle the module size and weights.

The framework's development is a collective research team effort. The main components that were done through this research are the visualization federates, visualization engines and their connections, other developments such as the simulation components of the framework are beyond the scope of this research. The visualization is concurrent with the simulation allowing the user to observe the simulation behavior and providing a better representation of spatial changes on a construction site and the logic of the construction process.

5. Two-way communication between the simulation and visualization components and framework time management

As previously stated, a major contribution of this research to the body of knowledge is that it will provide a framework necessary to enable two-way communication between simulation models and visualization components. The framework communications are done

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

through several layers. The visualization federates reflect the simulation's object classes attributes values as they are updated by the simulation federates. These reflections are communicated between the two federates through the RTI of the HLA configuration. On the other hand, the viewer federates and the 3D visualization engine(s) communicate through a network User Datagram Protocol (UDP), which provides language-independent communications between the .NET based HLA federation and 3D engine.

Both, the Site Visualization Federate and the BGE, act as clients and servers to send and receive messages between each other using a Python API. Different ports have to be assigned for sending and receiving per application. This communication has built on the strength already inherent in the distributed simulation HLA standards, by allowing the site visualization 3D engine component and the construction simulation component to physically work—exist on—different terminals connected through networks.

Operating the framework using federates which are distributed to different physical locations was already tested and proven successful. In this case, the connection takes place on different ports with different IP addresses instead of having all ports on the local host. The sending and receiving ports on each side are conceptually explained in Fig. 2 below.

Another interesting aspect of this framework is how the simulation time is synchronized between the simulation and visualization components of the framework. Time management inside the framework is organized through the HLA IEEE standards. This is mainly handled by identifying federates as time-regulating, time-constrained, or both. This means that the overall simulation time of the federation is controlled through time regulating federates which do not allow the global simulation time of the federation to advance unless they receive a “Time Advance Grant” from the RTI. The grant is sent once all the simulation events that should be processed before the requested time advancement (time the federation will advance to) are fired. On the other hand, time-constrained federates, such as the Visualization Federate(s); advance their times according to the messages from the Time Regulating Federates (Simulation Federates). Accordingly, the site Visualization federate sends the changing values of the object class attributes to the BGE to dynamically update site layout as per Fig. 2 and the BGE sends back the pathfinding mechanism results. A sample high level abstraction of the sequence of messages exchanged through the RTI during simulation is shown in Fig. 3.

6. Development of mobile resource pathfinding mechanism

The pathfinding mechanism is an integration of mesh generation mechanism and A* algorithm; used to determine the shortest and obstacle-free path and its expected mobilization time. It is also utilized by the visualization component to generate depiction of the site for each resource mobilization event. Fig. 4 explains the conceptual data exchange between the simulation and visualization components during the simulation run, together with the inputs and outputs of the pathfinding mechanism.

Initially, a complete 3D prototype of the construction site is produced inside the visualization component of the BGE before the simulation run starts. This prototype, which can be imported from AutoCAD 3D or others drafting packages, contains the initial layout of the project at time zero (before the resource mobilization phase commences). Also, it contains a 3D model of the facility and a 3D model repository of key mobile resources, which are either critical to perform project activities or expensive, in terms of time and cost, to mobilize.

In this initialization, the user assigns locations to the temporary facilities that change throughout the course of the project. The HSV framework enables changing and updating these locations as the simulation runs via the visualization screen. Once the simulation starts, the visualization component starts receiving updates from the Site Construction Federate; accordingly, depicts the various construction processes including key mobile resources and changes in the site layout. These updates result from changes in the project (progress), mobile resources (utilization or location), interaction between them or the simulation time. Fig. 5 illustrates the pathfinding mechanism; it explains the link between the mesh generation mechanism and the A* algorithm. Since it is at the top of the hierarchy, it is referred to as the pathfinding mechanism Parent Flowchart.

6.1. Mesh generation mechanism

The mesh generation algorithm can be utilized for covering 2D and 3D spaces. Managing tempo-spatial aspects of resource mobilization is only a 2D problem; therefore, the site's spatial data are defined by the x and y axes, and the site layout and its borders are defined in terms of the global coordinates on the axes. The center coordinates are calculated as the user inputs the number of nodes along each axis. The more nodes covering the site space in each direction, the more complicated the A* algorithm calculations will be for each resource movement. The external borders of the site space are then defined for the x and y coordinates through the following equations:

$$X_{min} = X_c - X_l/2 \tag{2}$$

$$X_{max} = X_c + X_l/2 \tag{3}$$

$$Y_{min} = Y_c - Y_l/2 \tag{4}$$

$$Y_{max} = Y_c + Y_l/2 \tag{5}$$

where X_{min} and Y_{min} are the minimum border coordinates on one side along the x and y axes while X_{max} and Y_{max} are the border coordinates for the other side. X_c and Y_c define the center coordinates parallel to the x and y axes and. X_l and Y_l represent the site plot dimensions along the x and y axes.

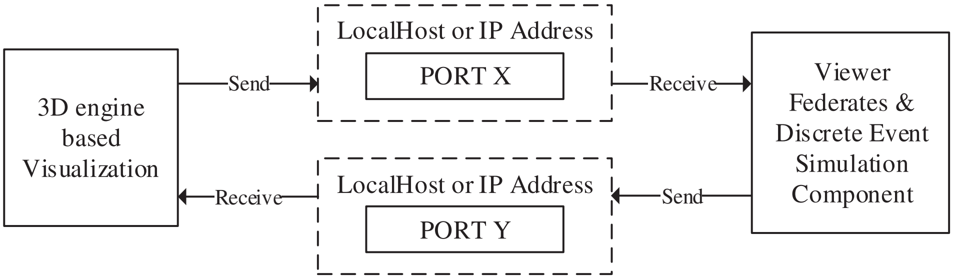


Fig. 2. Different ports on different workstations or on the same local host.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

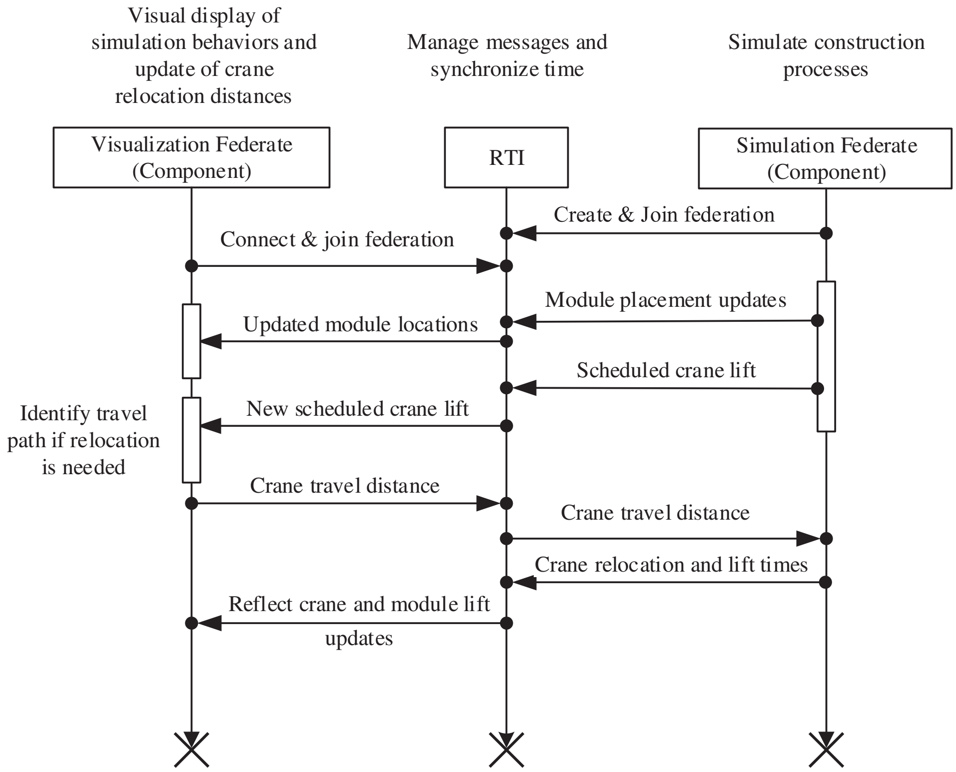


Fig. 3. High level of interactions between federates.

Once the site lot and the number of nodes along each axis are defined, the algorithm gets the coordinates of each node and convert it to [x,y,z] space coordinate node. Also, for each node, the adjacent nodes from the four directions are defined and stored. Mesh data are stored globally to facilitate its accessibility by other modules and functions within the HSV framework. Then, the site spaces that are occupied by temporary site facilities are subtracted from the initial mesh by using a ray cast function [56], built in the visualization component BGE. The function draws an imaginary ray from each node to its neighboring nodes. If the ray hits an object on the node itself or before it reaches the neighboring nodes, it means a structure, either permanent or temporary, is occupying this node. "Collision"; on the other hand, is a given property to any structure that occupies a site space. If there is a structure with this property, the site space status will change from a Resources Admissible Site Space (RASS) to a Resources Forbidden Site Space (RFSS), making it un-accessible by the A* algorithm. Fig. 6

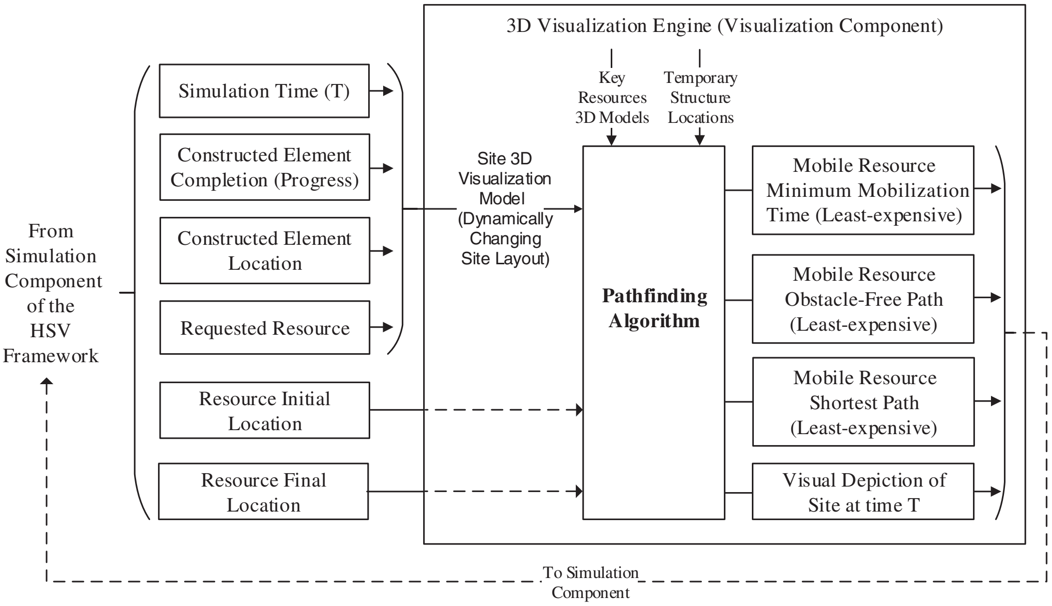


Fig. 4. Conceptual data exchange mechanism between simulation and visualization components.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

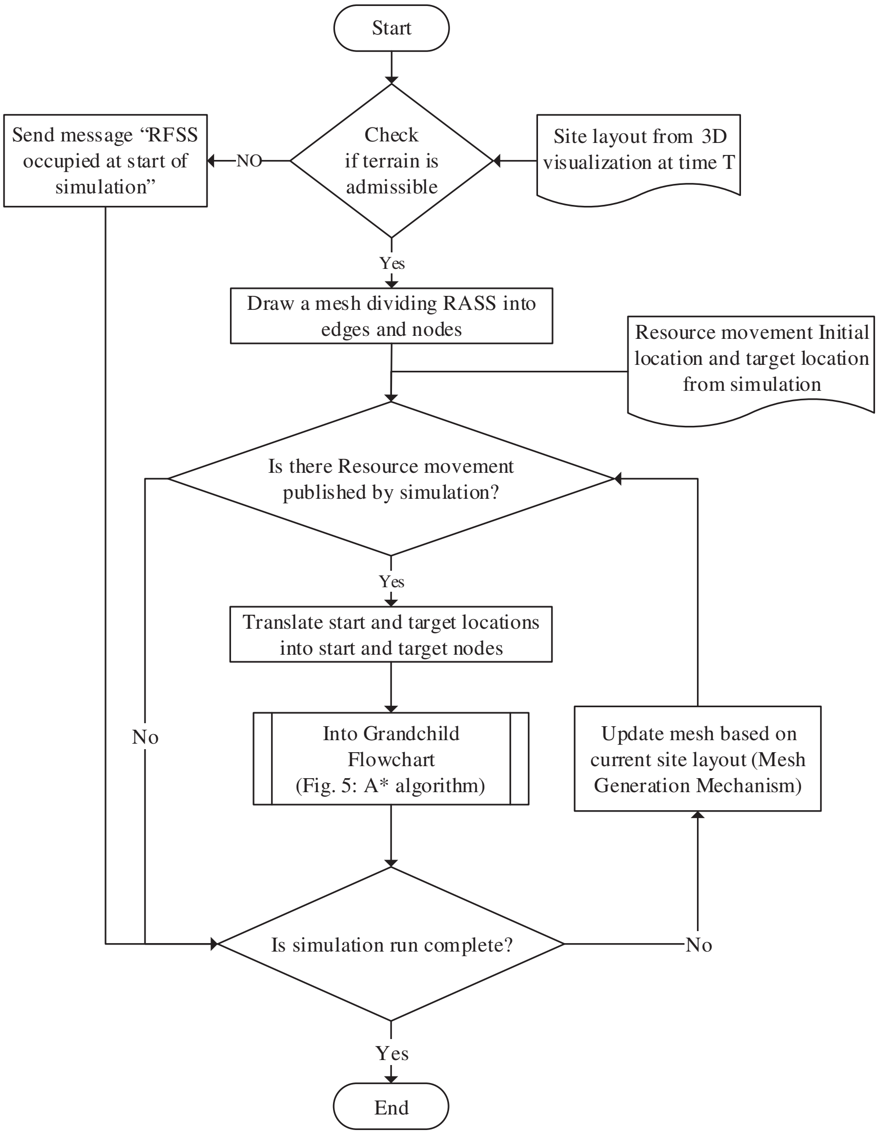


Fig. 5. HSV pathfinding process.

shows a generated mesh covering a site layout at simulation time zero. It shows the occupied areas by the footings and other representations of temporary structures are not covered by the grid. It also shows the pick-up locations for cranes (yellow markers) where specially designed mats can be placed to allow cranes to pick and place several modules from the same pick-up location. Crane movement between these locations to execute different lifts is triggered by the simulation component according to available modules.

The 3D component generates the initial mesh at simulation time zero based on the initial site layout and it covers the available site space for mobile resource mobilizations. As the simulation time advances from T(p) to the current simulation time (T), the project progresses and permanent structural elements evolve occupying site spaces that were marked as RASS. Concurrently, the mesh is updated to accommodate these changes on the site layout. The update is done through the BGE component through a customized mesh update function when site space is occupied.

Mobilizing and demobilizing temporary facilities; on the other hand, is not an event to be triggered by the Site Construction Simulation Federate. Temporary facilities are represented on the visualization screen by adjustable 3D blocks; in terms of dimensions and locations, that can be controlled automatically or manually by the user during the simulation. The user interacts with these blocks by simply moving them on the project plot using arrow keys. Every time the user changes the location of any of these blocks on the site plot, an update message will be triggered to recall the mesh update function; accordingly, updating the node mesh.

6.2. A* algorithm application

The A* algorithm uses the mesh created by the mesh generation algorithm to mark the RASS and choose the shortest path for a certain mobile resource. The nodes (vertices) represent the traveling points, while the edges represent the feasible paths between these nodes. During the simulation, if a resource mobilization event is scheduled, the A* algorithm goes through these nodes and edges to find the shortest path and yield the distance between a resource's origin and destination points (start and target nodes). It first adds the start node to a list called Closed List and then proceeds adding its adjacent nodes to a list called Open List. Based on the $f(N)$ evaluation function, the A* algorithm adds the node with the lowest travel cost to the Closed List making

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

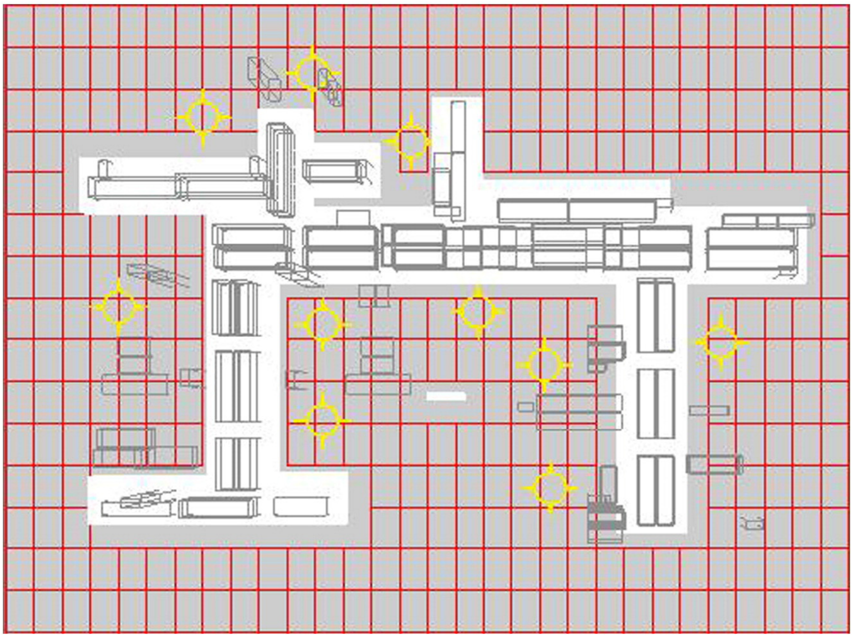


Fig. 6. Mesh covering resources admissible site space.

the start node its parent and rename it as current node. This process will be repeated for each current node added as follows:

- If its adjacent nodes are in the RFSS or in the Closed List, they will be ignored.
- If its adjacent nodes are not on the Open List, they will be added to the Open List and its $f(N)$, $g(N)$ and $h(N)$ costs will be calculated.
- If its adjacent nodes are already on the Open List, the algorithm calculates their travel costs, adds the lowest travel cost node to the Closed List and renames it as current node.

The process stops when the target node is reached and added to the Closed List, in which case the path has been found. Also, the process stops if the algorithm has gone through the maximum number of nodes identified by the user or when the Open List becomes empty, yet failing to reach the target node. In which case there is no feasible path found and adjustments; removing obstacles, to the current layout will be required. Then, the algorithm goes backward from the target node to its parent until the start node is reached marking the route, as green line, and sends back its information to the simulation. Illustration of the A* algorithm is presented in Fig. 7; it is the Grandchild Flowchart since its operation depends on the mesh generation mechanism.

6.3. Generation of visual behavior

Generation and customization of the behaviors of 3D objects is discrete in both the BGE and simulation and it is done using graphical logic bricks. These logic bricks are built-in constructs in the blender environment that allow developers to create interactive game environments. Objects equipped with these bricks can respond to different events happening in the environment or generated by user activities. The logic bricks are broken down into three main types: "sensors", "controllers", and "actuators". Sensors have different types and their main function is to sense a certain action or event that happens in the environment. Once a sensor is triggered, it sends a signal to its controller. Controllers can be connected to different sensors and act as logical gates (e.g. "AND", "OR") to actuators. When all conditions of a controller are met, it fires one or more actuators. Actuators produce the visual

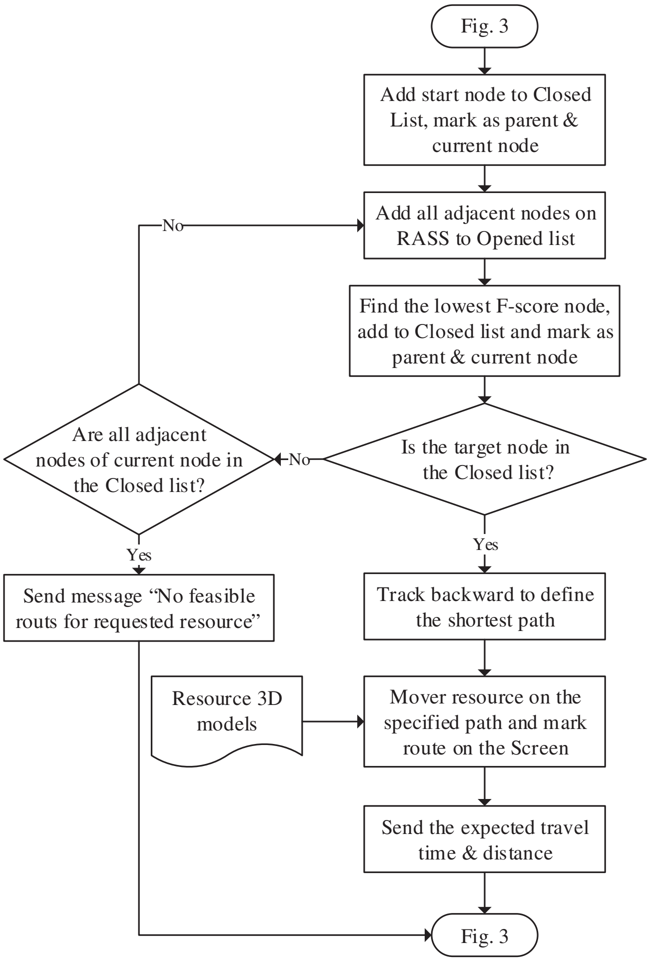
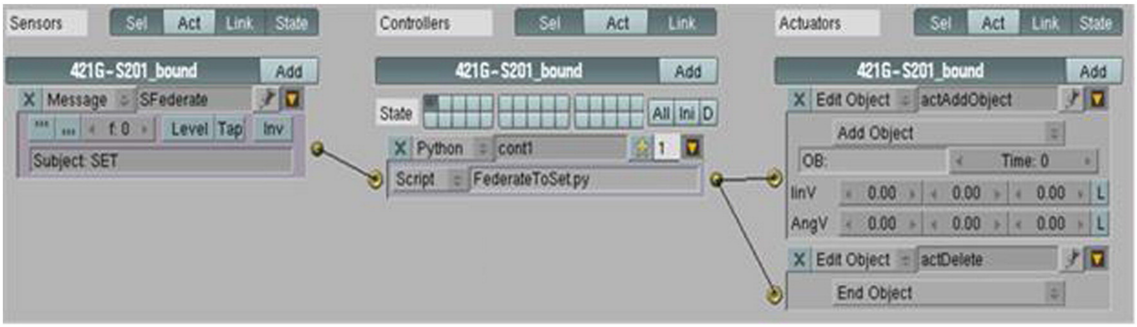


Fig. 7. A* pathfinding algorithm.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013



Message SFederate

Subject: SET

Python cont1

Script FederateToSet.py

Edit Object actAddObject

Add Object

Edit Object actDelete

End Object

Fig. 8. Logic bricks for displaying modules in site viewer.

behaviors in the 3D environment. A set of actuators owned by a certain 3D object will trigger it to produce different transformations according to the type of these actuators and the controller they are connected to.

As an example, a 3D environment is created for the site visualizer containing different 3D objects. Some objects are static and they are mainly to enhance realism of the environment. Dynamic objects are equipped with logic bricks to react to simulation messages. The dynamic objects inside the site visualization BGE are: 1) Module prototype models that represent 3D models of the actual modules, and 2) bound objects that represent markers of the final set points of the different modules and help manage the placement of module prototypes at their final locations on site, as shown in Fig. 8.

One part of the information generated is an attribute “ModuleFieldLocation” that describes a module final location (storage, pick point, or set point). If this attribute is updated, the visual behavior is then triggered. A sensor is created to listen to messages with a subject equal to the string “SET”. When such a message is sent to the object, it triggers two actuators. The first adds the prototype model of the module to the scene at its final location while the second deletes the bound object itself from the scene as it is no longer required.

A similar visual behavior is to represent the dynamic changes in mobile crane location. It is associated with updates in “CraneID” and “LocationID” attributes. Time-stamped messages are fired when a lift activity involving mobile crane is scheduled by the simulation federates. The actuator deletes the crane object from its current location once the “LocationID” attribute is updated, and adds it to the new location. The crane incremental movement is not simulated or displayed as crane locations are preset matching the practice on construction sites where crane locations and module pick points are determined prior to the start of the construction.

7. A sample case

The following example is extracted from running the framework to simulate and visualize heavy crane mobilizations on an industrial construction project. The project represents the construction of an upgrader for crude oil. The facility is located in the industrial development of Scotford, northeast of Ft. Saskatchewan, Alberta. The Scotford Upgrader has a rated processing capacity of 355 barrels per day (56,000 m^3/d) [57]. It is built using preassembled modules manufactured off-site in a fabrication shop and then shipped to the site and placed into position using mobile cranes. The industry research partner was responsible for the construction of the Residue Hydro Conversation (RHC) Integrated Hydro Treatment Unit with total of 140 modules delivered to the site [57]. This example demonstrates how the simulations, translations to visual behaviors, and the pathfinding mechanism interoperate to produce resource tempo-spatial behaviors.

The lifting event involves the mobilization of a crane from one lift location to another. The lift locations are predefined onsite for each set of modules. Fig. 9 shows the initial site layout at simulation time zero with the expected lift locations presented in yellow ring markers, excavation and footing areas presented in white, and small rectangulars representing temporary facilities. Once a lifting event is triggered, the

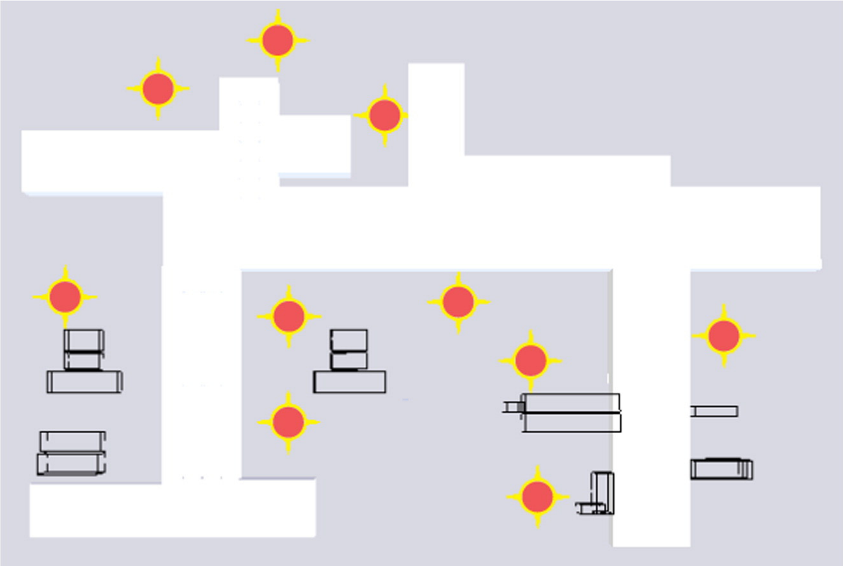


Fig. 9. Initial site layout.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

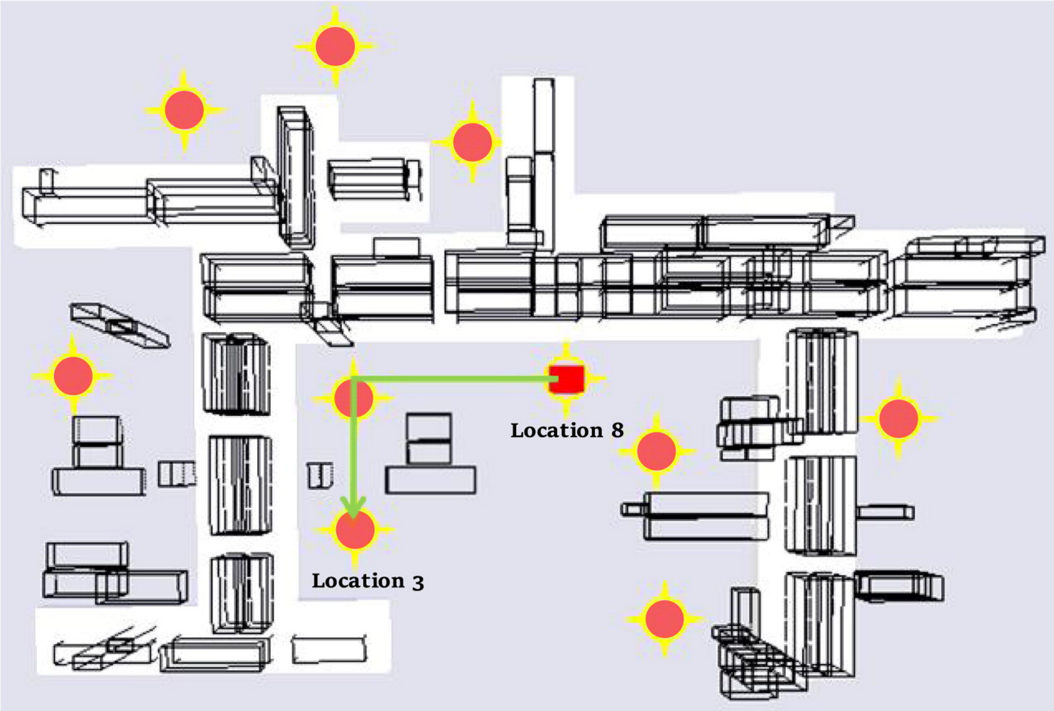


Fig. 10. Crane 2 traveling route. (For interpretation of the references to color in this figure, the reader is referred to the online version of this chapter.)

crane moves from its initial location to its targeted location, where the next group of modules is to be installed.

Concurrently as the simulation starts, the visualization component generates the first mesh to cover the initial site layout that is divided into RASS and RFSS. The Site Construction Simulation Federate starts sending messages to the visualization federate to be translated into visualization behavior and changes on the site layout; accordingly, updating the mesh covering the RASS. Simultaneously, the A* algorithm analyzes each mobilization event requested and sends a feedback to the simulation containing its expected mobilization time and distance, ID of the resource on the move, its original location, and its final mobilization location and displays the route it will take on the visualization screen.

As shown in Fig. 10, a message is sent by the Site Construction Simulation Federate to move crane 2; presented by the red cube, from location 8 to location 3 to start a lift operation. Based on this layout, the A* algorithm shows the shortest obstacle-free path; presented by the green line, with its expected distance of 351.4 m as the visualization

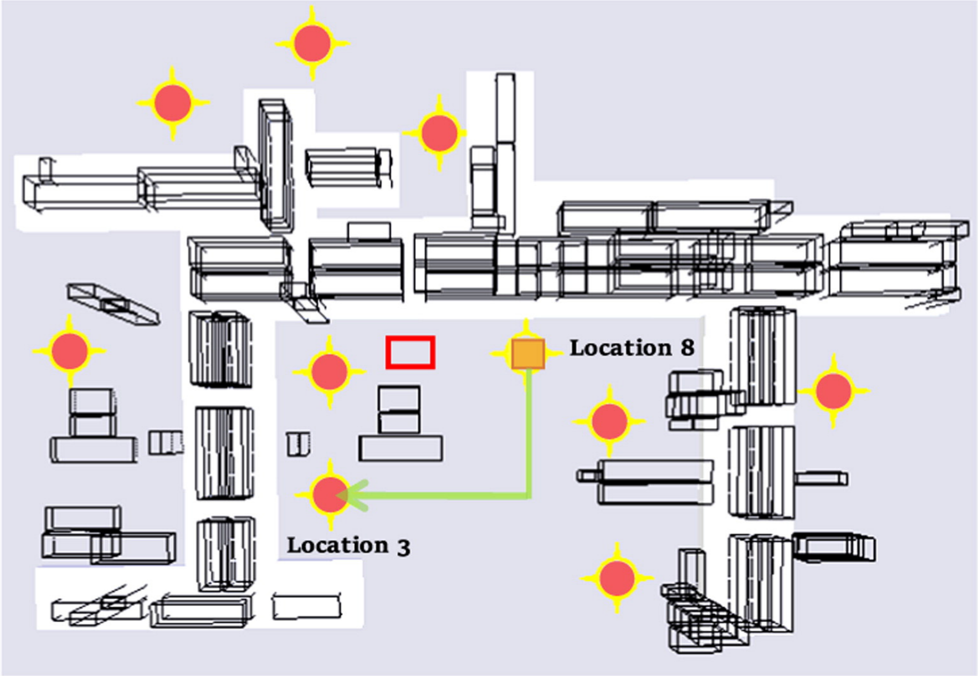


Fig. 11. Crane 4 traveling route. (For interpretation of the references to color in this figure, the reader is referred to the online version of this chapter.)

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

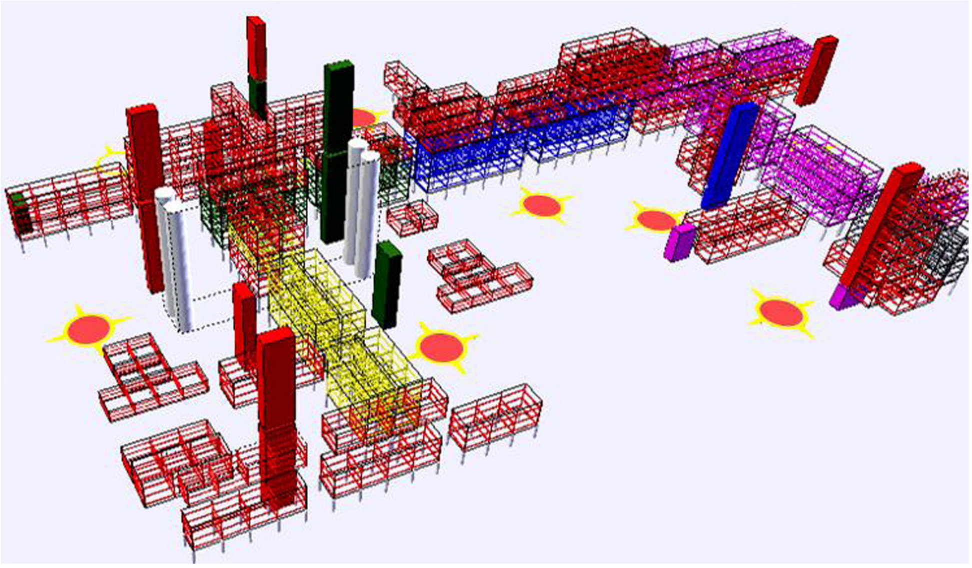


Fig. 12. Site after the simulation run.

component completes the movement and sends its feedback to the simulation.

The user can change the location and size of temporary construction facilities at any time during the simulation by moving or resizing the white rectangular boxes. To demonstrate, a temporary construction facility was placed on the chosen path between locations 8 and 3 (red triangle); as shown in Fig. 11. Once a module was added, the site space that was free became occupied by the new module changing the space state from RASS to RFSS. As the simulation runs, an event to move another crane (crane 4, orange cube) on the same path followed earlier was scheduled; however, there was a temporary structure blocking this path, changing part of the site space from RASS to RFSS. Therefore, crane 4 that almost has similar dimensions of crane 2 was given a different path by the A* algorithm to move between the same locations. Although, the extra travel of 30 m might not be a significant increase, it demonstrates the resource planning capabilities of the proposed framework.

Fig. 11 illustrates how the A* algorithm accounts for changes of installing permanent structures in the site layout as the simulation runs. At time T, when a movement event for the crane was scheduled, the A* algorithm identified its movement path. As the simulation progressed; reached time T + t, several modules (permanent structural elements) were installed and a new movement event was scheduled for the same locations, thus a different movement path is identified. Fig. 12 shows a screenshot of the visualization component after the end of the simulation run.

8. Model output evaluation

As a simplified alternative to the proposed HSV framework, simulation of travel times can be modeled using different probabilistic distributions to stochastically represent the variation in resource's traveling distance. In order to verify the effectiveness of the proposed HSV framework, a sample output after a complete run of the framework was modeled using Generalized Extreme Value Distribution. The theoretical and empirical Cumulative Distribution Functions (CDFs) were compared at a range of relatively low significance levels. Fig. 13 shows the sample data (generated from the framework) in comparison to the

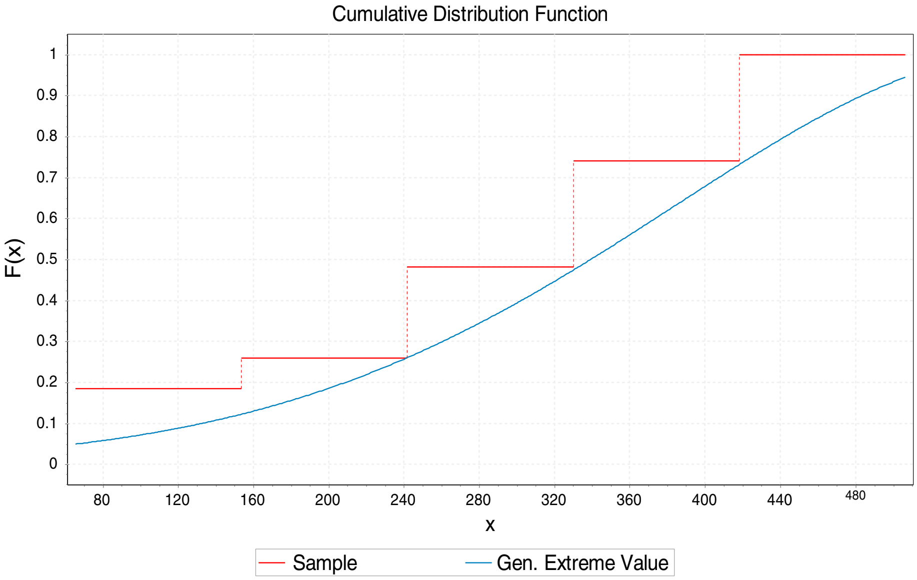


Fig. 13. Cumulative Distribution Functions of HSV sample data vs Generalized Extreme Value Distribution.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

fitted Generalized Extreme Value distribution. Fig. 14 (P–P plot) shows a comparison between the empirical CDF value points against their theoretical CDF values.

The plot shown in Fig. 14 should fall in the comparison line if the specified theoretical distribution is the correct model. Though, the Generalized Extreme Value Distribution is found to be the best fit, the deviation is clear. This indicates that probability distributions can be misleading as the shortest distance estimated here is not the average traveling distance of mobile resources onsite. If the travel distance is estimated randomly from a stochastic distribution, there is a high probability that the result will exceed the distance of the shortest path. On the other hand, a deterministic approach such as the pathfinding algorithm implemented in the HSV framework estimates the shortest distance for the least-expensive path rather than an average distance that is modeled based on past experience and observations.

Moreover, no construction sites have identical layouts; a stochastic distribution that is sampled from observations from one site will not be indicative of mobile resources' shortest paths in a different site. Most importantly, fitted distributions also do not reflect the temporal order between the generated values due to the dynamic changes in a site layout. In this case, earlier travel distances are likely to be shorter due to free site space than later distances where more space is occupied.

9. Limitations and future research

Identifying the optimal pathfinding algorithm is not the focus of this research; the scope is to demonstrate the benefits of integrating decoupled engines with a show case of path planning application. The Mesh Generation Mechanism and A* algorithm implemented in this framework have their limitations when searching for the mobile resources shortest paths. Alternative path finding algorithms can replace the A* algorithm in future work and can be evaluated for efficiency and impact on overall simulation behaviors. The mesh generation mechanism does not take into consideration the change in terrain that results from changes in site topography during its operation. A terrain with steep slope can hinder the mobilization of heavy lift mobile resources and should be represented as a RFSS. The Pathfinding Mechanism Extension implemented here does not consider such a space to be a RFSS. It also leaves a safety buffer zone marked as RFSS around permanent structures to assure that the shortest path chosen will not cause any collisions between mobilized resources. This method assumes that the traveling mobile resources have a uniform shaped footprint (for example square or rectangular). Also, it does not allow full collision detection when it comes to resources with irregular shaped footprints. Future research should address these limitations, for instance; the safety buffer zone should be adjusted based on the movable resource's footprint or bounding mesh. Also, the resource path problems can be treated as “Robot Navigation” problems, where the resource has an infinite rather than a discrete set of routes to move in a dynamic site layout.

Even though the two-way communication mechanism presented in this framework enables information exchange between the visualization components and the simulation, the current state of art imposes the limitation of real-time interaction between the simulation model and jobsite to reflect dynamic changes as the project evolve. Therefore, it is important to note that the proposed simulation model is not integrated with construction site and its source of input is a predefined graphical representation of the site which is embedded in the visualization component. Although the user can relocate objects in the visualization component testing different scenarios and obtain feedback accordingly, a mechanism for the simulation model to be adaptive to the dynamic of the construction site can be a valuable extension to this work. Future research should introduce methods to overcome this limitation, for instance, using mathematical algorithms to formulate random generators to capture construction data with realistic composition of attributes (e.g. weather generator). More specifically, capturing field data to verify and update the input parameters of the visualization component can ultimately contribute to the process of simulation modeling and developing more realist simulation driven visualization models.

10. Conclusion

This work presents enhancements for simulation-driven visualization. It proposed a solution to model mobile crane movements under tempo-spatial changes of construction site layout that result from either project progression or relocation of temporary facilities. The enhancement integrates discrete simulation capabilities with a 3D visualization component; exploiting their capabilities for time and resource representation and analysis. Two-way communication and integration between the two technologies are done through distributed simulation protocols.

The application involves an implementation of a pathfinding algorithm that integrates mesh generation and A* algorithm to model movements of mobile cranes. The pathfinding algorithm assesses

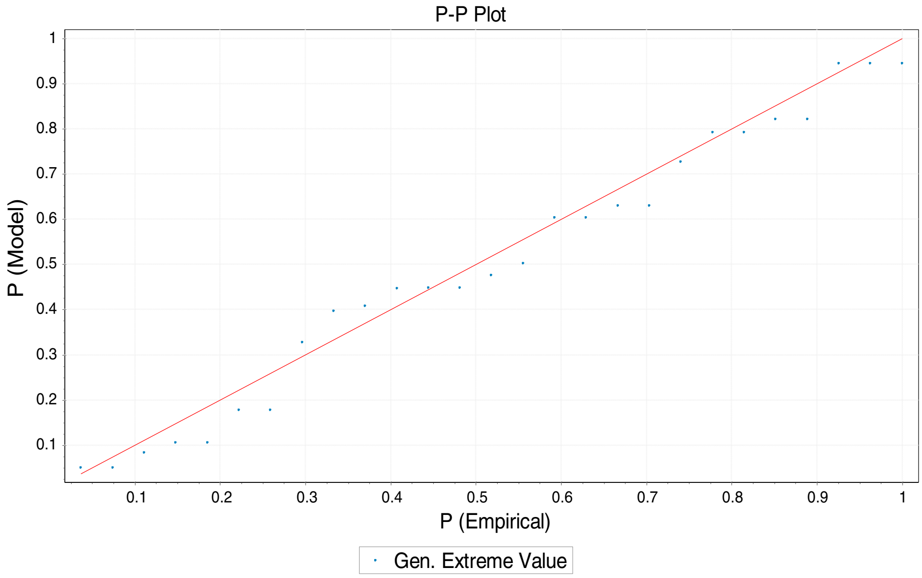


Fig. 14. Probability–probability plot of CDF values.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013

feasible paths for each mobilization event, determines the shortest obstacle-free path and calculates its associated duration to complete the event. The durations provided through path finding analysis provide a more accurate representation of time required to move the cranes and are then used by the discrete event simulation model. Demonstration of the framework capabilities was also presented through a prototype used on a real world industrial construction project. Limitations associated with the proposed framework are also discussed, for example, the mesh generation mechanism disregard to change in terrain during its operation as well as not enabling real-time integration of the simulation model with jobsite. Future research should introduce methods to overcome this limitation and advance the current state of art, importantly, by enabling real-time interaction between the simulation model and jobsite. The proposed framework will contribute towards the construction industry by integrating time related site space management with simulation capabilities.

Acknowledgment

The authors would like to acknowledge the contribution of the Natural Science and Engineering Research Council of Canada (NSERC) and collaborating construction companies for their funding of this research through the Collaborative Research and Development Grant number CRDPJ 335345-05.

References

[1] D. Halpin, R. Woodhead, Design of Construction and Process Operations, John Wiley and Sons, New York, N.Y., USA, 1976.
[2] A. Kalk, S. Douglas, Insight: interactive simulation of construction operations using graphical techniques, Technical Report No. 283, Construction Institution, Department of Civil Engineering, Stanford Univ., Stanford, California, USA, 1980.
[3] V. Kamat, J. Matrinez, Visualizing simulated construction operations in 3D, J. Comput. Civ. Eng. 15 (4) (2001) 329–337.
[4] R. Akbas, Geometry Based Modeling and Simulation of Construction ProcessesPh.D. thesis Dept. of Civil and Environmental Engineering, Stanford University, Palo Alto, CA, 2003.
[5] V. Kamat, J. Matrinez, Software mechanisms for extensible and scalable 3D visualization of construction operations, J. Adv. Eng. Softw. 39 (8) (2008) 659–675.
[6] S. AbouRizk, Role of simulation in construction engineering, J. Constr. Eng. Manag. (2010) 1140–1153.
[7] A. ElNimr, Y. Mohamed, A simulation driven visualization framework for construction operations: development and operation, Proc. Constr. Congr., Banff, 2010.
[8] A. ElNimr, Y. Mohamed, Application of gaming engines in simulation driven visualization of construction operations, Electron. J. Inf. Technol. Constr. 16 (2011) 23–38.
[9] K. El-Rayes, H. Said, Dynamic site layout planning using approximate dynamic programming, J. Comput. Civ. Eng. 23 (2009) 119–127.
[10] N. Dawood, R. Marasini, Visualization of stockyard layout simulator “Simstock”: a case study in precast concrete products industry, Autom. Constr. 12 (2003) 113–122.
[11] F. Zhou, S. Abourizk, H. Al-Battaineh, Optimisation of construction site layout using a hybrid simulation-based system, Simul. Model. Pract. Theory (2009) 348–363.
[12] M. Andayesh, F. Sadeghpour, Dynamic site layout planning through minimization of total potential energy, Autom. Constr. 31 (2013) 92–102.
[13] E. Elbeltagi, A.T. Hegazy, Eldosouky, dynamic layout of construction temporary facilities considering safety, J. Constr. Eng. Manag. 130 (2004) 534–541.
[14] J. Zhang, M. Anson, Q. Wang, A new 4D management approach to construction planning and site space utilization, 8th Inter. Conf. on Comput. Civ. Build. Eng., Stanford University California, USA, 2000.
[15] Z. Ma, Q. Shen, J. Zhang, Application of 4D for dynamic site layout and management of construction projects, Autom. Constr. 14 (2005) 369–381.
[16] F. Sadeghpour, O. Moselhi, S. Alkass, Computer aided site layout planning, J. Constr. Eng. Manag. 132 (2006) 143–151.
[17] I. Tommelein, R. Levitt, B. Hayes-Roth, SightPlan model for site layout, J. Constr. Eng. Manag. 118 (4) (1992) 749–766.
[18] X. Su, A.r. Andoh, H. Cai, J. Pan, A. Kandil, H.M. Said, GIS-based dynamic construction site material layout evaluation for building renovation projects, Autom. Constr. 27 (2012) 40–49.
[19] H. Said, K. El-Rayes, Optimal utilization of interior building spaces for material procurement and storage in congested construction sites, Autom. Constr. 31 (2013) 292–306.
[20] Y. Zhou, L. Ding, X. Wang, M. Truijens, H. Luo, Applicability of 4D modeling for resource allocation in mega liquefied natural gas plant construction, Autom. Constr. 50 (2015) 50–63.
[21] B. Choi, H. Lee, M. Park, Y. Cho, H. Kim, Framework for work-space planning using four-dimensional BIM in construction projects, J. Constr. Eng. Manag. 140 (9) (2014) 04014041.
[22] H. Said, K. El-Rayes, Automated multi-objective construction logistics optimization system, Autom. Constr. 43 (2014) 110–122.
[23] S.S. Kumar, J.C. Cheng, A BIM-based automated site layout planning framework for congested construction sites, Autom. Constr. 59 (2015) 24–37.
[24] F. Sadeghpour, O. Moselhi, S. Alkass, Modeling the time factor in site planning, Proc. Annu. Conf. Can. Soc. Civ. Eng., Toronto, 2005.
[25] T. Gominuka, F. Sadeghpour, A framework for activity-based identification of space requirements for construction equipment, Proc. Annu. Conf. Can. Soc. Civ. Eng., Quebec City, 2008.
[26] K. Tantisevi, B. Akinci, Automated generation of workspace requirements of mobile crane operation to support conflict detection, Autom. Constr. 16 (2007) 262–276.
[27] H. AlBahnassi, A. Hammad, C. Zhang, Accurate heavy equipment motion planning considering local and global constraints, Proc. Annu. Conf. Can. Soc. Civ. Eng., St. Johns, 2009.
[28] B. Paulson, S. Douglas, A. Kalk, A. Touran, G. Victor, Simulation and analysis of construction operations, J. Tech. Top. Civ. Eng. 109 (2) (August 1983) 1983.
[29] A. Oloufa, K. Crandall, Feedback mechanism for operational simulation, J. Comput. Civ. Eng. 6 (2) (April 1992) 1992.
[30] J. Xu, S. Abourizk, Product-based model representation for integrating 3D CAD with computer simulation, Proceedings of the Winter Simulation Conference, Phoenix, USA, 1999.
[31] Y. Zhang, M. Al-Batahineh, J. Kostelyk, S. AbouRizk, Practical application of 3D/4D modeling and process simulation in construction, Proc. 13th International Conference on Construction Applications of Virtual Reality, London, UK, 2013.
[32] T. Ni, H. Zhang, C. Yu, D. Zhao, S. Liu, Design of highly realistic virtual environment for excavator simulator, Comput. Electr. Eng. 39 (2013) 2112–2123.
[33] M. González, A. Luaces, D. Dopico, J. Cuadrado, A 3D physics-based hydraulic excavator simulator, Proceedings of the ASME/AFM 2009 World Conference on Innovative Virtual Reality WINVR2009, Chalon-sur-Saône, France, 2009.
[34] F. Vahdatikhaki, A. Hammad, Framework for near real-time simulation of earthmoving projects using location tracking technologies, Autom. Constr. 42 (2014) 50–67.
[35] F. Vahdatikhaki, A. Hammadb, H. Siddiqui, Optimization-based excavator pose estimation using real-time location systems, Autom. Constr. (2015) 76–92.
[36] A. Oloufa, Modelling and simulation of construction operations, Autom. Constr. 1 (4) (March 1993) 1993.
[37] G. Wainer, Q. Liu, Tools for graphical specification and visualization of DEVS models, SIMULATION 85 (3) (2009) 131–158.
[38] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, Academic Press, San Diego, CA, 2000.
[39] M. Ali, M. Fagiar, Y. Mohamed, AbouRizk, Beyond classic models — design and development of comprehensive earthmoving simulator, 14th International Conference on Construction Applications of Virtual Reality, Sharjah, UAE, 2014.
[40] A. ElNimr, Y. Mohamed, Loosely coupled visualization of industrial construction simulation using a game engine, Winter Simulation Conference, Phoenix, AZ, 2011.
[41] P. Rekapalli, J. Martinez, Runtime user interaction in concurrent simulation–animations of construction operations, J. Comput. Civ. Eng. 23 (2009) 372–383.
[42] P. Rekapalli, J. Martinez, Discrete-event simulation-based virtual reality environments for construction operations: technology introduction, J. Constr. Eng. Manag. 137 (2011) 214–224.
[43] A.H. Behzadan, C.C. Menassa, A.R. Pradhan, Enabling real time simulation of architecture, engineering, construction, and facility management (AEC/FM) systems: a review of formalism, model architecture, and data representation, J. Inf. Technol. Constr. (ITcon) 20 (2015) 1–23.
[44] L. Song, N.N. Eldin, Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling, Autom. Constr. 27 (2012) 32–39.
[45] R. Akhavian, A. Behzadan, Automated knowledge discovery and data-driven simulation model generation of construction operations, 2013 Winter Simulation Conference (WSC), 2013.
[46] R. Akhavian, A. Behzadan, Knowledge-based simulation modeling of construction fleet operations using multi-process data mining, J. Constr. Eng. Manag. 139 (11) (2013) 04013021.
[47] R. Akhavian, A.H. Behzadan, Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers, Adv. Eng. Inform. 29 (2015) 867–877.
[48] Blender Foundation, Help and education, 2010 [Online]. Available: http://www.blender.org/education-help/ (Accessed 10 Jun. 2010).
[49] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, second ed. Prentice-Hall, New Jersey, 2003.
[50] L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Trans. Robot. Autom. 12 (4) (1996) 566–580.
[51] J. Barraquand, J.C. Latombe, Robot motion planning: a distributed representation approach, Int. J. Robot. Res. 10 (1991) 628–649.
[52] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. SSC4 4 (2) (1968) 100–107.
[53] IEEE, 1516, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules, IEEE, New York, NY, USA, 2010.
[54] IEEE, 1516.1, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification, IEEE, New York, NY, USA, 2010.
[55] IEEE, 1516.2, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification, IEEE, New York, NY, USA, 2010.
[56] S.D.R. Roth, Casting for modeling solids, Comput. Graphics Image Process. 18 (2) (1982) 109–144.
[57] PCL Constructors Inc., Shell RHC area construction works, 2015 [Online]. Available: http://www.pcl.com/Projects-that-Inspire/Pages/Shell-RHC-Area-Construction-Works.aspx.

Please cite this article as: A. ElNimr, et al., Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changin..., Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.05.013