

COMPARISON OF MANUAL AND AUTOMATED SIMULATION GENERATION APPROACHES AND THEIR USE FOR CONSTRUCTION APPLICATIONS

Gunnar Lucko

Department of Civil Engineering
620 Michigan Avenue NE
Catholic University of America
Washington, DC 20064, USA

Perakath C. Benjamin
Kannan Swaminathan

Knowledge Based Systems, Inc.
1408 University Drive East
College Station, TX 77840, USA

Michael G. Madden

M. Madden Consulting, LLC
2570 Red Maple Place
Melbourne, FL 32935, USA

ABSTRACT

This paper compares two fundamentally different approaches and their efforts to create functional simulation models to analyze and optimize construction operations. It contrasts the traditional manually created discrete-event simulation with an automated simulation model generation engine. While input data remain the same for both, the former requires a user to extensively determine, create, and connect the different elements, followed by an often time-consuming verification to correct flaws in details. The latter has the proven potential to radically reduce the time, cost, and skills of creating complex models by using process templates from which models for construction applications can be rapidly deployed. Moving from the traditional paradigm to automated, yet user-supervised modeling can finally make the rich body of knowledge in simulation accessible to practitioners, who can reap new benefits from being able to rehearse their projects in the computer and optimize their processes before any costly physical resources are committed.

1 INTRODUCTION

Computer-supported process simulation has been explored for several decades in the academic world (Oldfather, Ginsberg, and Markowitz 1966) and has a rich history of developing a series of increasingly sophisticated simulation environments that were designed for construction applications (e.g. Martínez 1996, Liu 1991, Halpin 1973), while often being suitable for general-purpose applications as well. The major advantages of using simulation prior to actual processes in the physical world have been listed by Lucko et al. (2009, p. 2733) as their “controlled environment”, safety, speed, custom focus on “the phenomenon of interest” only, ability to perform “sensitivity analysis”, repeatability to enable probabilistic and Monte Carlo-type analyses, and “allow experimentation with large and complex systems with randomized behavior.” These many advantages are specifically useful to the practice of construction management, which must plan and control the complex processes to put into place built facilities and systems that support the modern quality of life.

Additionally, the cost of a full simulation study, whether self-performed or by a simulation consultant, including the license for a commercial software package, may still be significantly lower than for holding a large-scale experiment with actual physical resources, e.g., an earthmoving process with trucks and loaders. In fact, aside from small static mock-up of intricate assemblies, e.g. part of a brick wall with a pattern bond, construction projects hardly ever make use of rehearsal-type optimization of its processes.

1.1 Challenges of Traditional Approach to Simulation Modeling

All discrete-event simulation models are based on the premise that the computer handles the interactions between its basic elements, activities and resources, at specific points in time whenever any one of them incurs an event (Lucko et al. 2008). For activities the event is a state change from idle to active and *vice versa*; for resources it is the acquisition or release of any unit thereof by or from a queue that holds them. An activity will check if all of its starting conditions are fulfilled (in the form of available resources and finished predecessor activities) and commence if all these constraints return ‘true’ values. Different from scheduling, however, simulation can efficiently model any repetitive processes with its cyclic structure.

However, as is described in more detail below, creating a simulation model manually element-by-element for any construction application is time-consuming and requires a high degree of familiarity with the behavior of the simulation elements at the lowest possible level of detail. Furthermore, it may require a training comparable to a graduate education course in simulation methodology before the user can successfully deploy a model that not only functions correctly but is also adequate for the desired analysis.

Detriments that appear to have prevented the ubiquitous use of simulation as an everyday tool in the practice of construction management have been cited exhaustively in the literature as the *time, cost, and skills* (e.g., Benjamin, Patki, and Mayer 2006; Cates et al. 2002; Son and Wysk 2001) that it takes to sufficiently master – in that order – both the theory of simulation and the practice of its specific software implementation. An alarming disconnect between academia and practice in regard to simulation is the fact that of the various existing software implementations that have been developed by researchers over the past three decades, not a single one has found its way to commercialization beyond licenses for not-for-profit educational and research purposes. In other words, even while the new generation of construction managers has learned simulation, typically in a graduate course, once they enter practice there is little if any routine use of these valuable tools. In fact, it has to compete with static and discrete scheduling, primarily through the widely used critical path method (Galloway 2006) that offers none of the advantages and analytical capabilities mentioned in the previous section. This paper therefore explores how the nature of simulation modeling itself is executed traditionally, with few exceptions, and how it measures up to an automated approach.

1.2 Template-Based Simulation Environments

A few simulation environments have attempted to ease the effort of creating simulation models by introducing template-based systems. *Process templates* “are reusable building blocks that can be quickly combined and edited to form different workloads” (Lucko et al. 2009, p. 2738). While their use has been criticized as self-limiting because “diverse and complex nature of construction processes will always require a function of a form not foreseen by the system designer (or requiring a choice of operands or operators not provided in the template)” (Martínez 1996, p. 409 f.), it is still possible to model even novel techniques (Lucko Benjamin, and Madden 2008) by creating various *general patterns of operations* that encompass even unforeseen novel techniques that, after all, must still be composed of activities and resources as their basic elements.

Process templates thus group clusters of elements into reusable model fragments from which a new overall model can be created rapidly. Previous attempts to use templates in simulation of construction processes were made by Odeh (1992), who provided a library of construction methods and hoped that users would expand it over time, Shi (1995), who similarly built a resource-based library of only earthmov-

ing process, AbouRizk and Mohamed (2000), who developed a framework for composing special-purpose models and performing various analyses, and Lu et al. (2007), who flow-sequenced clusters of activities.

They can overcome the painstaking ‘puzzle’ of assembling many elements at the lowest possible level of detail – activities and resources – and replace it with a ‘modularization’ akin to the prefabrication of complete assemblies in the building industry that merely need to be selected, transported, and connected.

The following sections review in detail the individual steps that a simulation modeler has to complete to produce a functional and valid simulation model of a construction process. They are derived from practical and educational experience of the authors. Due to the uniqueness of each simulation modeling effort, they can merely provide a general chronological flow of the challenges that must be typically overcome.

2 DETAILED TRADITIONAL APPROACH

Simulation is only a surrogate for actual experimentation (Law and Kelton 1991). Any models must be developed based on the best available observation and understanding of the real-world system. The following sections describe the detailed steps that the traditional approach takes. A ‘good’ simulation model that is created through careful application of these steps is one that is capable of providing all of the information to complete the desired analysis while being parsimonious in its use of modeling elements for a number of reasons: To save *time, cost, and effort* in developing the model structure and input, executing it on a computer, and analyzing its output, to allow easy verification, and to support successful presentation to its future users who accredit it (Lucko and Rojas 2010) as credible and reliable for use in daily practice.

Yet beyond introductory tutorials on specific software and its modeling conventions (Martínez 1998) most of the attention in scholarly articles has been given to describing the real process and an analysis with its representation in model, but *not* how said model itself was built. For example, a recent thorough simulation study of concrete paving for a highway project to evaluate the impact of factors (Hassan and Gruber 2008) omits the actual user-performed modeling entirely and skips directly from a process narrative to a description of the completed cyclic simulation network. Obviously, the essential question of *how* simulation models are best created for successful use by practitioners has thus far largely escaped the attention of researchers but was eclipsed by *what* analytical capabilities the plethora of systems provided.

2.1 Problem Definition

Just as any scientific research must identify a clear research question, so does a simulation study have to first define the actual problem that must be solved, finding a balance in optimizing the time-cost performance of a construction process. An unclear description of what exact question the user wishes to answer will lead any subsequent modeling and analysis off target and its results may be verified, but never valid.

2.2 Process Narrative

The second step in building a model is to carefully assemble and understand all of the available information about the process and its product into a process narrative. Often this requires extended conversations between the simulation modeler and the subject matter expert. Once a draft has been composed it is good practice to validate it with the future user or beneficiary of the model that is being developed. This written document is also called an assumptions document, as it not only contains all of the known information in verbal or numerical form, but also any assumptions that need to be made about the process, i.e. to resolve ties with case distinctions, and ideally gives justification for their values. Good engineering judgment, technical expertise, and common sense that are required in this step make traditional simulation modeling both science and art. The importance of this narrative is underlined by the fact that all of the quantitative and qualitative input for the simulation model will be extracted from it in the following steps. An example of such narrative for a multi-story commercial building project has been provided by Lucko et al. (2009).

2.3 Model Specifications

The next step is to determine what resolution (i.e., granularity or level of detail) the model needs to have. For example, shall only truckloads full of soil be modeled, or each bucketful, or every cubic meter? Shall every brick be modeled, or each square meter of wall, or only the entire wall? The resolution directly influences the run time of the simulation model on a computer as well as the depth of the analysis that can be performed from its output, i.e., the quality of interpretative statements. Furthermore, it must be determined which elements are essential to the process and will be modeled (e.g., excavators, trucks, and their operators) and which elements are non-essential and are not modeled (e.g., office staff). To set the proper scope or complexity (Chwif, Barretto, and Paul 2000) one should follow the general principle of Ockham's razor, named after a medieval monk, to make the model as complex as necessary but as simple as possible. A hierarchical structure is beneficial here because it allows adding or removing detail at a later point in time.

2.4 Activity and Resource List

The next step extracts mentioned or merely implied modeling elements, i.e. activities, resources, and their relationships from the narrative. Each of these three elements must be refined into values and conditions:

- Activities have a name, possible states, and a duration (a probabilistic distribution if applicable);
- Queues are created to hold resources at some expected minimum, maximum, or average amounts;
- Relationship links have conditions of preceding elements and transmit specific units of resources.

The thinking approach used here is similar to the well-known work breakdown structure (WBS) of cost estimating. As when using a WBS, a consistent labeling convention is kept for giving each modeling element its unique identifier, which is required internally by the simulation engine and also for readability of small icons in the graphical representation of the simulation network, e.g. 'LdTrk' for an truck loading activity. It is good practice to write all three element types in tabular form, which assists in fully and correctly capturing their data. This is similar, yet more comprehensive, to the activity list that schedulers use.

2.5 Draft Activity Cycle Diagram

Using a paper and pencil, sketches of the simulation network are created in successively more complete and correct iterations. Such activity cycle diagram (Tocher 1961) is similar to a flowchart in that it includes conditions. In the authors' experience this traditional tool is very valuable, while more experienced simulation modelers may directly work in the computer. Still, converting the activity and resource list into a simulation network requires care, for example, 'dummy' elements become necessary to model complex constraints such as clocks for uptime and downtime, traffic routing and control for earthmoving (Ioannou and Martínez 1996), or safety regulations for quarrying (Martínez and Ioannou 1995). For clarity, the simulation network should group sub-processes into clusters. For readability, it should direct the overall process from top left to bottom right and clockwise for any cyclic loops. It is good practice to align its elements in the rows and columns of an invisible grid, which are connected in paths that are as direct as possible with minimum crossovers; the latter is extremely complex to solve mathematically and requires trial-and-error.

2.6 Add Programming Commands in Simulation Language

As mentioned, modeling elements can theoretically be used to model complex constraints by serving as non-physical 'dummy' elements that control certain aspects of the process and issue signals on its states, for larger models it becomes more efficient to exploit object-oriented capabilities that modern simulation engines offer, e.g. the Stroboscope language (Martínez 1996), to define reusable variables and operators.

2.7 Verification (Debugging)

Experience shows that, depending on the complexity of the model and experience of its modeler, it may take roughly about as long to verify its computer file as it already took to draft it with paper and pencil. While the number of possible errors is infinite, they can be categorized and sequenced in their typical order of appearance to assist in efficient and successful ‘debugging’ a model before further alpha testing:

- Checking syntax conventions, e.g., some systems do not allow blank spaces in names or – a very common error – their programming code requires an end-of-command marker, e.g., a semi-colon;
- Checking names, duration distributions, and conditions of all elements for typographical errors;
- Using an alternating sequence of activities and resources as per the particular simulation engine;
- Following each branch of the activity cycle diagram for completeness and correctness compared to the process narrative and the elements and conditions of the tabular activity and resource list;
- Checking the consistency of time units throughout the model, which typically is entirely unitless;
- Checking the integrity of resources that are transmitted through links. Their numerical values may denote different entities, e.g., when 10 cubic meters plus 1 truck merge to become 1 loaded truck;
- Checking if inequality (<, >) operators of IF-THEN starting conditions on links become fulfilled;
- Dividing the narrative into parts and drafting each as a model fragment (template), whose inputs and outputs are then linked, e.g., the ‘load-haul-dump-return cycle’ and similar building blocks;
- Rewording complex conditions to better understand their nature or modeling their inverse, e.g., changing “add soil to storage” to the functionally equivalent “subtract space from storage”;
- Identifying non-physical constraints and signals that can be modeled by using ‘dummy’ elements.

The actual behavior of the simulation engine can give important clues as to the type of error, e.g., if the model fails to execute it may be the case that none of the overall starting conditions are fulfilled. It is good practice to follow the event trace or log files for at least two full cycles of the process to survey both the warm-up phase and the steady-state execution of the model. An animation interface helps visualizing the process as modeled for comparing it with the real-world scenario and identifying irregularities.

With the aforementioned steps completed the model should be functioning correctly for an analysis.

3 AUTOMATED SIMULATION APPROACH

3.1 Knowledge Based Modeling Approach

This paper is philosophically grounded in a *knowledge based approach* that focuses attention on the preliminary aspects of the simulation process, beginning with acquisition and a system description, followed by its analysis, and culminating in the design of a conceptual simulation model. Such approach is by definition *description-centered*. When asked to explain a problem, domain experts often find it convenient to communicate by relating an ordered sequence of activities that describes ‘how things work’ in their domain. Much of the knowledge to reason about the problem can thus be found within the description itself. To support the discovery process for building idealized models from such unstructured descriptions, one must have methods and tools to acquire, represent, and reason descriptive knowledge (Lucko, Benjamin, and Madden 2008).

3.1.1 Knowledge Engineering and the IDEF Modeling Languages

Decades of research in the area of knowledge and systems engineering (Mayer, Painter, and deWitte 1992) led to design of “the Integrated Modeling Definition (IDEF) family of methods for various applications in data structuring and analysis. It has its origins in the business and military environment, but is in the public domain as a neutral symbolic system of semantics and syntax... IDEF has the advantage of being a well-established approach whose various methods are fully integrated with each other, including IDEF3 for process flow and object state description capture, IDEF4 for object-oriented design, and IDEF5

for ontology description capture” (Lucko, Benjamin, and Madden 2008, p. 2482). More details on its capabilities are provided in these sources.

3.1.2 Overview of the WorkSim[®] Knowledge Based Planning and Scheduling System

WorkSim[®], KBSI’s knowledge based planning and scheduling system, cost effectively addresses planning, scheduling, and process analysis needs of large, complex organizations (Lucko et al. 2009). It manages different types of process information including task precedence constraints, priorities, durations, resource availability schedules, etc. WorkSim[®] uses advanced search and simulation optimization methods to determine optimal planning. Advanced simulation-based experimentation capabilities facilitate sensitivity and ‘what if’ analyses. WorkSim[®] schedules can be exported in multiple formats to, e.g., Microsoft[®] Project and Microsoft[®] Excel. Project management tools can then perform further schedule analyses such as CPM/PERT and resource leveling, and provide alternative visualizations of the schedule like Gantt bar charts and PERT Diagrams. Reports allow focusing on specific aspects of plans to identify areas for improving processes. The WorkSim[®] knowledge based approach allows users to quickly make changes to process information in response to changes in real world system requirements. Once they are made, users can simply rerun the WorkSim[®] scheduling engine and quickly generate a revised schedule that propagates implications of changes through the model and effectively shows their impact on the overall plan.

It has been successfully used in the aerospace industry (Benjamin et al. 2005; Benjamin, Graul, and Erraguntla 2002). However, it is important to note that the use of simulation has not yet reached critical mass within the construction industry in spite of preliminary initiatives in this area (Lucko et al. 2009; Lucko, Benjamin, and Madden 2008).

3.2 Initial Model Design and Development Steps

An important and early step is to understand and capture a description of the problem from the user perspective in construction management. An example problem description motivates the simulation model development process. A manager of a contractor considers bidding for a large multi-building construction project. Based on past project data they would like to know the estimated average resource utilization and construction time (with 90% confidence) and which resources will likely become ‘bottlenecks’ to make contingency arrangements for them. Construction involves a cluster of 5 buildings that shares many similar characteristics. The start of construction for the different buildings is staggered by 3 months. The manager would like to use simulation to answer the following questions about the optimal use of resources:

- Do we have enough capacity for this contract? Which resources are scarce or have extra capacity?
- Assuming a triangular (95%, 125%) duration variability, a bonus for finishing early, and a penalty for finishing late, which specific resources should be acquired more of, and how many of these exactly?

Note that using simulation to solve real-world problems is an iterative and collaborative approach, requiring several information exchanges between simulation analysts and the subject matter experts (SME). Other important stakeholders also participate in the modeling and analysis process, e.g. managers who often are the consumers and use the resulting information for decision making. The early problem definition step is often neglected in practice, resulting in models that are designed to solve wrong or non-existent problems. Consequently, an incorrect problem definition will lead to waste of scarce time and money and will reduce confidence of the efficacy of simulation as a decision support tool for real-world applications.

3.3 Acquire and Merge Existing Data

A difficult task is to identify, understand, and collect input data for the simulation model. In the example case study (Lucko et al. 2009), pre-existing data were identified at the contractor. However, they were not in any form that was immediately suitable for modeling that would address the target problem. Multiple steps were required to transform these available data into the appropriate form for simulation modeling:

- 1) *Identify data sources*: It is not uncommon to incur a scenario where the data are splintered across different data systems in an organization. They may also be stored in different file formats such as scheduling software files, spreadsheet workbooks, design drawings, and even text documents.
- 2) *Collect data*: Not all data required for the simulation model may be readily available in electronic format. The modeler will have to interview the SME to collect required data. In the case study, an interview with the scheduler gave handwritten resource requirements that were later transcribed.
- 3) *Develop understanding of data*: Reviewing all of the files for format, contents, and completeness.
- 4) *Merge data into usable form*: Since data files are ‘owned’ by different individuals, there may not be a common ‘key’ that allows for easy merging of data. The modeler may therefore need extract, transform and load (ETL) tools to repair, format, and merge the data. In this case study, the authors were able to merge and format their input data using standard spreadsheet functionalities.

Data in the case study were in two formats, Microsoft® Project (task names, durations, and dependency converted from the contractor’s Primavera files) and Microsoft® Excel (transcribed resource names, capacities, and tasks). They were merged to produce coherent data for building the simulation model. An important and non-trivial step in the data merge was to find unique identifiers that established dependencies between tasks and the resources needed to accomplish them as shown in Figure 1. Data were imported into an Excel file and formatted to comply with WorkSim® representation guidelines. This ensures consistent specification of different process constraints, e.g., task duration formulas, multiple types of precedence constraints, multiple types of task and resource dependencies, calendar constraints, and so forth. At this point, an important step was to verify the integrity of the imported data as a valid conversion of the raw data to help ensure that the future model would be sound and adequate to address its goals.

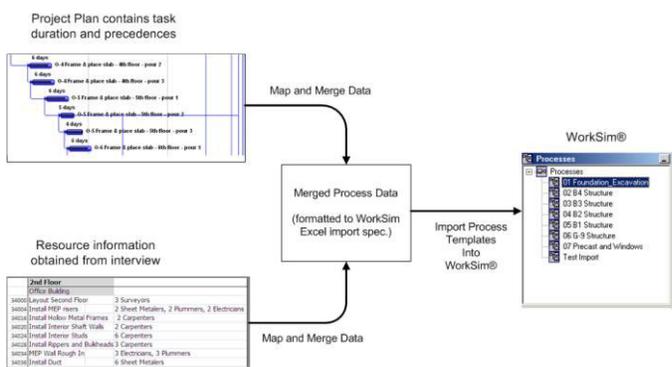


Figure 1: Mapping and merging different data sources.

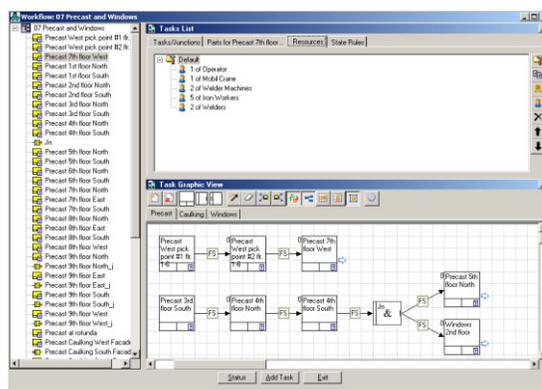


Figure 2: WorkSim® process editor.

3.4 Simulation Model Development

3.4.1 Create Process Templates

Process templates facilitate effective model management, maintainability, and reuse. They represented recurring groups of task in the case study, e.g. mobilization, foundation and excavation, basement structure, ground floor structure, above ground structure, precast installation and window fitting, basement finishes, above ground finishes, elevators, etc. A model is composed by selecting and ‘stitching’ templates together in the proper sequence. WorkSim® offers a comprehensive process editor to specify the tasks of a workload as shown in Figure 2. Process steps are refined in this interface, i.e. adding tasks and ‘junctions’ (inter-process constraint modeling artifacts), editing them, and modifying the flow and hierarchy of tasks. One can also build resource and state rules for tasks, assign attributes to tasks and junctions, and assign specific work teams to tasks. Users can navigate through their model, create and save diagrams, and make changes with the graphical user interface (GUI). Diagrams focus attention to certain key sections of the

process. For example, when working with a process model of thousands of tasks, it is impossible to present the entire flow in single diagram. Instead, the user can create logical sub-sections of the process such as disassembly, inspection, repair, installation, final inspection, etc., and then analyze them individually.

3.4.2 Define Resources, Skills, and Calendars

Resources are agents that are required to accomplish a task. WorkSim[®] offers four primary types: Personnel, equipment, skill, and location. Two additional categories, generic and virtual, are provided to create *logical* resources to model certain real-world phenomena. A common use of virtual resources is modeling of workspace constraint which limits the number of resources that could work concurrently in such area.

Resources have one or more skills. This allows for constructs such as *Bob* is an *Operator* and an *Electrical Technician*. Note that resources and skills have a many-to-many relationship. Skill ratings specify relative preferences between resources, and can be a resource selection policy. If multiple resources are available for a task, the one with higher skill rating will typically be selected by the simulation engine.

Users input the availability or exceptions of a resource in a GUI of a weekly work calendar. One can specify that *Bob* is available on weekdays from 8am to 5pm, except on January 1, July 4, November 24, and December 25. Exception calendars are crucial to assess the impact of non-availability of one or more resources on flowtime, e.g., if *Bob* attends a training program. The modeler first generates a baseline schedule assuming availability and then compares it with the *Item Flowtime* report of a second simulation run whose workload is re-scheduled subject to all of the appropriate calendar exceptions of the resources.

3.4.3 Specify Task Durations

Variabilities associated with task durations are modeled using duration ‘formulae’. If historical data exists for probability distributions, graphical plots and goodness-of-fit tests verify how well a distribution represents reality (Law 2008). Once its parameters are determined and verified, the script editor interface in WorkSim[®] is used to develop and test duration formulae. It supports several theoretical distributions and checks the syntax of the duration script. Duration data in the case study were optimistic estimates. Tasks that occur early during construction, e.g. excavation, foundation, concrete, steel erection, façades with windows, precast, curtain wall, roofing, and site work are subject to weather delays that may extend the activity duration by 25%. If conditions are perfect, tasks may be completed in 95% of the specified time.

3.4.4 Define Workload Items and Specify Induction Dates and Priorities

In the case study, one workload item was created for each building. An item can be formulated as an instance of a process template, so that it inherits the task network, resource rules, duration formulae, and other constraints from that template so that the modeler can apply a general process flow design to several workload instances, e.g. multiple buildings in a residential development. Once instantiated, each workload can be customized. If needed, additional tasks, junctions, and diagram levels are added by either creating new elements in the process editor interface or by copying them from other workloads or other process templates. Finally the induction (i.e. start) date and priority of each workload item is specified.

3.4.5 Design and Configure Simulation Experiment

It is important to design an ‘experiment’ to systematically run the simulation model. This must be done so that output data will provide the information needed to address the simulation goals, e.g., answer the questions of the decision maker. Simulation experiment design involves strategic and tactical activities, including setting the number and length of runs, random number generation process, simulation parameters, and policies. Task selection rules govern how tasks are prioritized. For the case study, these rules were used:

Prioritize partial tasks: Ongoing tasks take precedence over new tasks; *item induction date*: Tasks are prioritized by induction date, i.e., earlier start dates receive priority over later ones (i.e. ‘FIFO’); *largest number of successors first*: If several tasks could be activated at one point in simulated time, the task with the most successors is selected to enable more parallel tasks and possibly reduce the flowtime; *shortest processing time first*; *earliest schedulable start time*: If a task was delayed due to resource constraints it has priority over new ones; *percent complete*: Gives priority to tasks with higher percent complete status.

3.4.6 Conduct Test Runs and Perform Model Verification

Model verification is an important model development activity. Useful model verification guidelines are:

Verify that task duration is the actual time needed to perform it, i.e., touch time. Many legacy systems record lapse time that includes time during which a task was waiting for resources to become available. Conduct a walkthrough of the process model with the SME using process diagrams. The diagram editor in WorkSim[®] simplifies this by automatically linking tasks as they are dragged and dropped into the diagram canvas. For complex templates with hundreds of tasks one should create a process diagram for each subject area; Tasks may preclude performing other tasks, or tasks may be performed only if the workload has a particular state. Complex timing requirements and other state conditions are modeled with state constraints in WorkSim[®]. Certain tasks are performed only in 3rd shifts or on weekends. Virtual resources are used to impose additional calendar constraints on tasks; Resources in an enterprise are often organized into groups and business rules disallow sharing between them. To model this, resources are grouped into different teams with team constraints on workload items; Virtual resources can also model workspace constraints that limit the number of people working in an area; Verify that all items have correct start dates. This setting imposes a ‘start-no-earlier-than’ constraint on all tasks of that item, even if the resources are available to perform them; *Flowtime* reports provide quick insights into the model performance during test runs. If the projected flowtime is much smaller than expected the most likely reason is the absence of precedence constraints on key tasks. Another likely reason is the omission of resource rules on key tasks that need one or more scarce resources. If the projected flowtime is unexpectedly large, verify the resource constraints. This error may result from an incorrect resource quantity or calendar.

3.4.7 Execute Simulation

Finally, options for trace and log files are set to execute the model and generate output. In the case study, the contractor stated that they could obtain more laborers as needed and the quantity constraint on *Laborers* was relaxed in the model. This feature is helpful when performing resource requirement planning. The *Resource Rollup* report indicates the resources required during any given period. Five workload items were included with the early start of these items being spread over a four-and-a-half month period.

3.4.8 Analyze and Verify Model Output

The last activity is to study the output and interpret its results. Outputs generated from the simulation execution are saved to a database. WorkSim[®] creates reports including *Item Flowtime*, *Parts Requirement*, *Utilization Summary*, *Utilization Pivot Charts*, *Resource Assignment*, and an *Executive Summary*. A test simulation run was performed with a scope of only one building. With all resource constraints turned off, the average flowtime was about 152 days, which was compared to the source data for model verification.

The authors chose to estimate the flow time and its variability for completing the foundation and excavation part of the construction process. As Figure 3 shows, foundation and excavation for building one have an average flowtime of about 170 days. The other buildings have a longer flowtime due to resource constraints with the fifth one taking about 210 days. Figure 4 shows how adding resources can significantly reduce the flowtime. As explained earlier, WorkSim[®] considers variability of task durations to estimate the variability of project. This *Resource Requirement* report clearly indicates months when demand

exceeds availability (see Figures 5 and 6). It can be adjusted to review longer time horizons, e.g., quarterly rollups, if desired.

Flowtime (in Days)				
Item Name	Type	Start Date	Earliest End Date	Avg Days
Bldg1 Foundation_Excavation	None	01-Jun-10	09-Nov-10	170.2
Bldg2 Foundation_Excavation	None	15-Jul-10	28-Dec-10	187.8
Bldg3 Foundation_Excavation	None	15-Jul-10	11-Jan-11	198.1
Bldg4 Foundation_Excavation	None	13-Sep-10	15-Mar-11	189.2
Bldg5 Foundation_Excavation	None	22-Sep-10	13-Apr-11	210.5

Figure 3: Flowtime report of initial model.

Flowtime (in Days)				
Item Name	Type	Start Date	Earliest End Date	Avg Days
Bldg1 Foundation_Excavation	None	01-Jun-10	02-Nov-10	158.6
Bldg2 Foundation_Excavation	None	15-Jul-10	21-Dec-10	171.7
Bldg3 Foundation_Excavation	None	15-Jul-10	23-Dec-10	174.5
Bldg4 Foundation_Excavation	None	16-Aug-10	22-Feb-11	196
Bldg5 Foundation_Excavation	None	15-Sep-10	24-Feb-11	182.2

Figure 4: Flowtime report with extra resources.

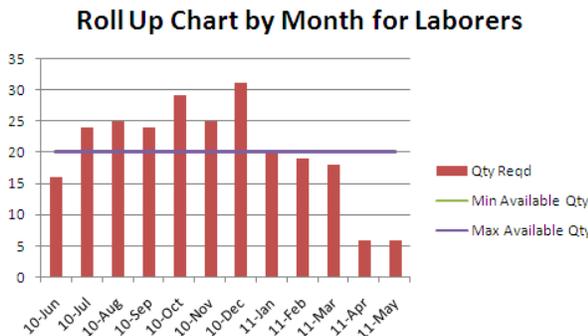


Figure 5: Initial resource requirements.

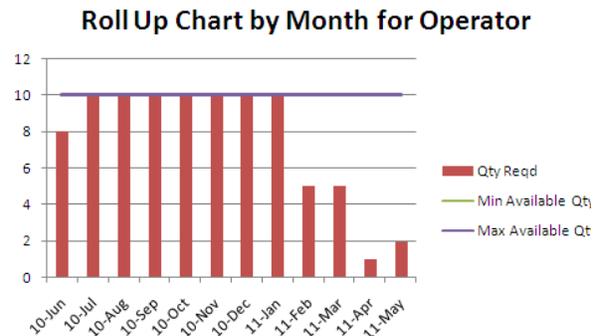


Figure 6: Improved resource requirements.

3.4.9 Refine Model and Re-run Simulation

The initial analysis of the output information revealed that the resources ‘Soil Mac’ and ‘Tieback Driller’ appeared to have excess capacity. On the other hand, ‘Operator’ and ‘Concrete Crusher’ were identified to be the main ‘bottlenecks’ (i.e. incur high utilization levels). If one wanted to assess the impact of having one additional Concrete Crusher and five additional Operators, one could create a copy of the existing resource profile in WorkSim®, increase their available quantities, and re-run the model to generate a new *Flowtime* report. The re-run output showed that here a significant reduction of process flowtime could be achieved for the different construction projects by increasing the availability of various scarce resources.

3.4.10 Interpret Results for Decision Making

The final activity in the simulation modeling and analysis process is to interpret its results to provide useful insights for decision making. In the case study, *Utilization* and *Flowtime* reports provided the information needed to answer the original questions posed by the decision maker (i.e. utilization, bottlenecks, flowtime, and flowtime variability). As described in the previous section, information about bottleneck resources was used to change resource availability levels. This mimics a real-world situation where bottleneck utilization levels are unacceptable to management. Alleviating the bottleneck induces a reduction in flowtime. Flowtime variability (interpreted as the degree of confidence in meeting the estimated project completion time with a given set of finite resources) can assist in managing the customer expectations.

Other project management functions can be explored by simulation. For example, if a given resource constraint is relaxed, another resource often emerges as a new bottleneck. In the case study, relaxing the constraints on the Concrete Crusher and the Operators resulted in the unintended effect of significantly increasing the utilization level of the resource ‘Backhoe’ as shown in the histograms of Figures 7 and 8.

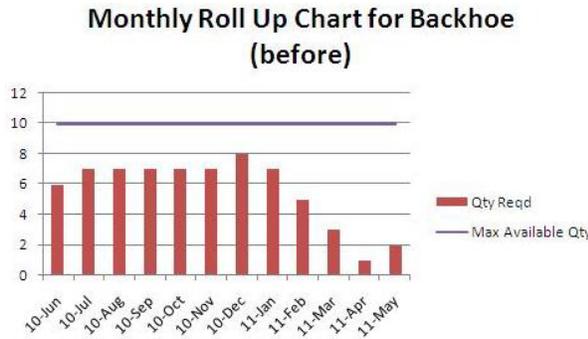


Figure 7: Initial resource histogram.

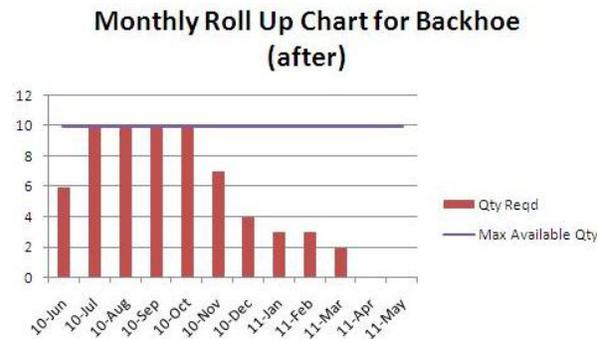


Figure 8: Unintended new bottleneck resource.

4 COMPARISON OF MODELING EFFORT

Simulation is an evolving science and practice. Previous tools, widely used in construction research and education, have enabled modeling, analyzing, and optimizing complex operations where a closed form solution for project flowtime estimation and resource requirement levels was not available. Advancement of computing technology has led to increasingly complex processes being successfully modeled. However, while simulation software evolves, more and more specific domain knowledge is required to produce quality simulation models. Whereas e.g. the manufacturing and aerospace industries are successfully utilizing complex simulation models, project management in the construction industry has yet to become aware of the full potential of process simulation and exploit its vast and valuable analytical capabilities.

Besides the aforementioned *time, cost, and skills*, another reason for this shortfall may lie in the nature of construction projects, which are considered ‘one-of-a-kind’ and may not have attracted sufficient practical interest in using simulation to improve and streamline their processes that are indeed repetitive.

This paper has attempted to provide an incentive and encouragement into this direction by providing a ‘whitepaper’ that compares the traditional approach of simulation modeling – not used in the construction practice – to existing and emerging capabilities of automated simulation generation. Extracting these features may lead to a domain-to-domain inspiration to more broadly share the benefits of simulation. In the long term, it may even be possible to create a translator system that is truly domain independent. IDEF is a highly promising approach into this direction. This could further make simulation directly available to construction managers without a lengthy learning curve. More research is needed to investigate exactly how simulation can be made more accessible to new domains while minimizing the required time, cost, and skills of the SME themselves. There is an urgent need to develop *measures* of time, cost, and skills so that the comparison of traditional, i.e., manual, versus automated simulation modeling can be made quantifiable, which can then assist in convincing the construction industry of the aforementioned rich benefits.

In closing, it is noted that simulation is ultimately a technique that helps gain insights and a better understanding of complex real-world phenomena. Consequently, the incremental results of a simulation project often triggers a new set of questions outside the scope of the original simulation study. Simulation results provide new information to the project stakeholders that can enable them to consider additional or expanded what-if scenarios and help them in addressing vital business problems. A flexible, knowledge based approach is therefore well suited for effective simulation modeling and analysis. As described in this paper, the simulation modeling process is time- and effort-intensive, highly dependent on SME expertise, and expensive. The typical lack of data that are already in a proper format ready to be used for a simulation model further compounds the problem. The authors contend that a knowledge based automation approach that facilitates cost effective adaptation and reuse will help address this problem. In particular, the repeated re-use and adaptation of process knowledge templates provides a key advantage. In the case study, the structured knowledge (organized as process templates) about a given building construction project can be efficiently adapted and reused for other construction projects. A well-engineered set of

templates in WorkSim[®] enables modelers to quickly adapt and change simulations for responding to continuously changing customer needs, i.e., it provides ‘agile’ simulation modeling and analysis capabilities.

ACKNOWLEDGEMENT

The authors wish to express their gratitude to the anonymous construction company that provided the data for this study and to their managerial staff who kindly shared their time and answered many questions.

REFERENCES

- AbouRizk, S. M., and Y. Mohamed. 2000. Symphony – an integrated environment for construction simulation. *Proceedings of 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1907-1914. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Benjamin, P. C., M. Graul, and M. Erraguntla. 2002. Toolkit for enabling adaptive modeling and simulation (TEAMS). *Proceedings of 2002 Winter Simulation Conference*, ed. E. Yücesan, C.H. Chen, J.L. Snowdon, and J.M. Charnes, 763-771. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Benjamin, P.C., K. V. Akella, K. Malek, and R. Fernandes. 2005. An ontology-driven framework for process-oriented applications. *Proceedings of 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2355-2363. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Benjamin, P.C., M. Patki, and R. Mayer. 2006. Using ontologies for simulation modeling. *Proceedings of 2006 Winter Simulation Conference*, ed. L.R. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, and R.M. Fujimoto, 1151-1159. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Cates, G. R., M. J. Steele, M. Mollaghasemi, and G. Rabadi. 2002. Modeling the space shuttle. *Proceedings of 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H., Chen, J. L., Snowdon, J. M. Charnes, 754-762. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Chwif, L., M. R. P. Barretto, and R. J. Paul. 2000. On simulation model complexity. *Proceedings of 2000 Winter Simulation Conference*, ed. J. A. Joines, Barton, R. R., Kang, K., Fishwick, P. A., 754-762. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Galloway, P. D. 2006. Survey of the construction industry relative to the use of CPM scheduling for construction projects. *J. Constr. Eng. Mgmt.* 132(7):697-711.
- Halpin, D. W. 1973. *An investigation of the use of simulation networks for modeling construction operations*. Ph.D. Thesis, University of Illinois, Urbana-Champaign, IL.
- Hassan, M. M., and S. Gruber. 2008. Simulation of concrete paving operations on Interstate-74. *J. Constr. Eng. Mgmt.* 134(1):2-9.
- Ioannou, P. G., and J. C. Martínez. 1996. Simulation of complex construction processes. *Proceedings of 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, 1321-1328. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Law, A.M., and W.D. Kelton. 1991. *Simulation modeling & analysis*. 2nd ed. New York, NY: McGraw-Hill.
- Law A. M. 2008. How to build valid and credible simulation models. *Proceedings of 2008 Winter Simulation Conference*, ed. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 39-47. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Liu, L.-Y. 1991. *COOPS – construction object-oriented process simulation system*. Ph.D. Thesis, University of Michigan, Ann Arbor, MI.
- Lu, M., W.-H. Chan, J.-P. Zhang, and M. Cao. 2007. Generic process mapping and simulation methodology for integrating site layout and operations planning in construction. *J. Comp. Civ. Eng.* 21(6):453-462.

- Lucko, G., P. C. Benjamin, and M. G. Madden. 2008. Harnessing the power of simulation in the project management / decision support aspects of the construction industry. *Proceedings of 2008 Winter Simulation Conference*, ed. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 2479-2487. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Lucko, G., K. Swaminathan, P. C. Benjamin, and M. G. Madden. 2009. Rapid deployment of simulation models for building construction applications. *Proceedings of 2009 Winter Simulation Conference*, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 2733-2744. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Lucko, G., and E. M. Rojas. 2010. Research validation in the construction domain: Challenges and opportunities. *J. Constr. Eng. Mgmt.* 136(1):127-135.
- Martínez, J. C. 1996. *STROBOSCOPE: State and resource based simulation of construction processes*. Ph.D. Thesis, University of Michigan, Ann Arbor, MI.
- Martínez, J.C., and P. G. Ioannou. 1995. Advantages of the activity scanning approach in the modeling of complex construction processes. *Proceedings of 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman, 1024-1031. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Martínez, J.C. 1998. EZStrobe – General-purpose simulation system based on activity cycle diagrams. *Proceedings of 1998 Winter Simulation Conference*, ed. D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan, 341-348. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Mayer, R. J., M. K. Painter, and P. S. deWitte. 1992. IDEF family of methods for concurrent engineering and business re-engineering applications. Report, College Station, TX: Knowledge Based Systems Available via <<http://www.idef.com/pdf/IDEFFAMI.pdf>> [accessed June 13, 2010].
- Odeh, A. M. 1992. *CIPROS: Knowledge-based construction integrated project and process planning simulation system*. Ph.D. Thesis, University of Michigan, Ann Arbor, MI.
- Oldfather, P. M., A. S. Ginsberg, and H. M. Markowitz. 1966. *Programming by questionnaire: How to construct a program generator*. Rand Report RM-5129-PR, Santa Monica, CA: Rand Corporation.
- Sawhney, A. 1994. *Simulation-based planning for construction*. Ph.D. Thesis, University of Alberta, Canada.
- Shi, J. 1995. *Automated modeling and optimization for construction simulation*. Ph.D. Thesis, University of Alberta, Canada.
- Son, Y.J., and R. A. Wysk. 2001. Automatic simulation model generation for simulation-based, real-time shop floor control. *Computers in Industry* 45(3):291-308.
- Tocher, K. D. 1961. The role of models in operational research. *J. Royal Stat. Soc.* 124(2):121-142.

AUTHOR BIOGRAPHIES

GUNNAR LUCKO is Assistant Professor of Civil Engineering at Catholic University of America. His research interests are modeling, analysis, and optimization of project schedules, equipment economics, and engineering education. He is a member of ASCE and INFORMS. His email is <lucko@cua.edu>.

PERAKATH C. BENJAMIN is Vice President for Research and Development at KBSI. He manages and participates in analysis, design, and development of advanced solutions for government and corporate clients. He has consulted on simulation and semantic technology. His email is <pbenjamin@kbsi.com>.

KANNAN SWAMINATHAN is Senior Systems Analyst at KBSI. He leads developing software systems to analyze, plan, schedule, and dispatch resources in process environments. Previously, he was a principal consultant on custom applications and eBusiness. His email is <kswaminathan@kbsi.com>.

MICHAEL G. MADDEN is President and Senior Consultant at M Madden Consulting LLC, Ph.D. Student at University of Miami, and former Senior Vehicle Engineer for space shuttles Endeavour and Atlantis. He is member of IEEE, AIAA, and INFORMS. His email is <mmadden.consulting@gmail.com>.