



Laval (Greater Montreal)

June 12 - 15, 2019

INTEGRATED SCHEDULING OF MODULAR CONSTRUCTION USING “MOD-SCHEDULER”

Salama, T.^{1,4}, Salah, A.², and Moselhi, O.³

¹ Department of Construction Management, California State University, Sacramento, California, USA.

² Department of Civil and Construction Engineering, College of Engineering, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam, Kingdom of Saudi Arabia.

³ Department of Building, Civil and Environmental Engineering, Concordia University, Montréal, Canada.

⁴ Salama@csus.edu

Abstract: Offsite construction has gained momentum in recent years due to its improved performance in projects' schedule, quality, safety, and environmental impact without increasing cost. Several research studies have introduced planning and scheduling techniques for modular and offsite construction using simulation-based methods. Few of these methods integrate offsite and onsite construction activities, and they are tailored to the specific needs of the project or the production system being considered. This process requires dedicated simulation professionals and historical records of productivity data which might not be available. Modular and offsite construction needs planning and scheduling methods that account for the unique characteristics of this industry such as its repetitive nature as well as uncertainty and variability of productivity rates. This paper presents a scheduling software prototype named “Mod-Scheduler”; designed as a web application for scheduling this class of projects. It enables integration of offsite and onsite activities for modular and offsite construction. Schedule development is based on buffering model presented earlier by the authors, which integrates linear scheduling method (LSM) and the critical chain project management (CCPM) while considering uncertainties associated with productivity rates. “Mod-Scheduler” was developed as a three-tier application tool comprising of 1) web front-end as the presentation layer coded using Angular, JavaScript (JS), hypertext mark-up (HTML) and cascading style sheets (CSS) languages; 2) ASP.NET core back-end system written in C# for data processing that includes calculations for activity durations, integration of offsite and onsite schedules by identifying the critical controlling link between both schedules, identification of overall critical sequence, and adding feeding and project buffers according to CCPM, and 3) SQL server database to store project data used in scheduling. Conclusions are drawn from using developed software to a case example to demonstrate its use and to highlight its capabilities.

Keywords

Integration, Scheduling, Modular Construction, Linear scheduling method, Critical chain project management, Software.

INTRODUCTION

Offsite construction and prefabrication facilitate work on offsite and onsite deliverables to reduce project schedule considerably. Reduction in schedule leads to cost savings due to a reduced need for expensive and labor intensive onsite operations. McGraw-Hill's Smart Market Report (McGraw-Hill, 2011) found that

67% of firms reduced project schedules using prefabrication and modularization and 35% of firms experienced schedule reduction of four weeks or more. Offsite construction provides valuable assistance for tight scheduling where project deadlines are strict as the higher education buildings sector (McGraw-Hill, 2011). Two features of modular and offsite construction control the determination of its appropriate scheduling techniques. The first is the repetitive nature of manufacturing for offsite construction, and the second is the uncertainty and variability of productivity rates for these operations. Thus, a traditional scheduling technique such as CPM is not the optimum scheduling method for this industry because it does not satisfy the resource continuity constraint required for repetitive projects (Harris and Ioannou, 1998). Simulation engines are used for scheduling modular and offsite construction due to their capabilities to schedule repetitive activities and uncertainty. However, simulation models require dedicated simulation professionals (Hajjar and AbouRizk, 2002) and are tailored to special needs and requirements of the project being considered. Hence, there is a need to develop a comprehensive resource driven methodology and automated software for scheduling of modular and offsite construction that accounts for uncertainty of productivity rates.

1 LITERATURE REVIEW

Parallel scheduling for offsite and onsite construction schedules saves 30 to 50% of project duration when comparing to stick-built traditional construction processes (MBI, 2016). Many studies have used simulation to schedule offsite construction processes (Abu Hammad (2003), Xie et al. (2011), Taghaddos et al. (2012), Ajweh (2014), and Altaf et al. (2015)), while other studies (Moghadam (2013), Liu et al. (2015), and Liu et al. (2016)) have integrated simulation with BIM to automate quantity take-off data and to visualize offsite and modular construction activities. Furthermore, linking the simulation model of the module yard (onsite) to the simulation model of the fabrication shop (offsite) is the main limitation of simulation models where the scheduler has to link both schedules manually (Taghaddos et al., 2009). Bu Hamdan et al. (2015) presented a system that integrates simulation with BIM techniques to provide an inventory planning and management tool while considering the interactions between different construction phases from manufacturing to assembly. However, this system has a less user-friendly graphic user interface (GUI), which makes it difficult for those who are not familiar with simulation tools. Moreover, the probabilistic simulation is fed manually by distribution functions.

Simulation is a good tool for probabilistic problems because the simulation model assesses the effect of uncertainties in construction projects (Lutz, 1990; Yang 2002). However, simulation cannot eliminate or control idle times by itself (Yang, 2002; Ioannou and Likhitrungsilp, 2005). Hence, an external algorithm must be implemented in the simulation model to solve the problem of repetitive scheduling effectively (Lutz 1990; Yang 2002). Without developing and implementing an external algorithm into simulation, simulation gives the same results as CPM, where activities are scheduled at their early start dates, while interruptions exist between units (Srisuwanrat, 2009).

On the other hand, linear scheduling maintains resource continuity and allows work interruptions in time-cost optimization problem. Analyzing and balancing the trade-off between work interruptions and continuity achieves minimum project cost by relaxing resource continuity constraints (Srisuwanrat, 2009). However, the LOB presentation was criticized by Arditi and Albulak (1986) and it was suggested that colored graphics be used to assist schedulers in differentiating between overlapping activities that have equal rates of production. LSM was also criticized for its inability to show task relationships and parallel activities graphically (Francis and Miresco, 2013). Mansur (1990) presented an approach that allocates critical activities on the upper side of the LOB chart and non-critical ones on the lower part for better presentation of LOB charts. Hegazy et al. (1993) introduced the BAL system as an extension to Mansur's work, which depends on an algorithm for scheduling and control of linear projects with an enhanced presentation to avoid activities overlapping. Salama et al. (2017a) extended to Mansur's work by presenting an alternative BIM based framework that integrates linear schedules of onsite and offsite construction operations using the integration between the linear scheduling method (LSM) and the critical chain project management (CCPM) in a manner that synchronizes work progress of these operations (Salama et al. (2017b)). This paper is presenting a scheduling software named "Mod-scheduler", programmed specifically for modular and offsite construction using ASP.NET system coded with C# programming language. This software integrates offsite and onsite schedules while considering uncertainty of productivity rates for both schedules and uses the integration method between offsite and

onsite schedules previously presented by Salama et al. (2017b). “Mod-scheduler” provides better visualization for the integration between offsite and onsite schedules and it is easy to use for any scheduler.

2 DEVELOPED METHOD

“Mod-Scheduler” was developed to schedule offsite and onsite activities for modular and offsite construction. Calculations for this software were based on the scheduling and buffering model introduced earlier (Salama et al., 2017a & 2017b) which integrates LSM and CCPM to schedule repetitive projects while considering uncertainty of productivity rates. The overall schedule was composed of an upper schedule for offsite activities and lower schedule for onsite activities, as shown in Figure 1. This overall schedule was generated after five sequential steps were followed: 1) calculation of aggressive and safe activity durations for offsite and onsite activities, 2) sequencing activities based on aggressive durations, 3) integration of offsite and onsite schedules by identifying the critical controlling link between both schedules (critical controlling link is shown as a green line connecting offsite and onsite schedules, as shown in Figure 1), identification of overall critical sequence (shown as the bold black line in Figure 1), and 5) adding feeding and project buffers (shown as blue and red lines, respectively, in Figure 1).

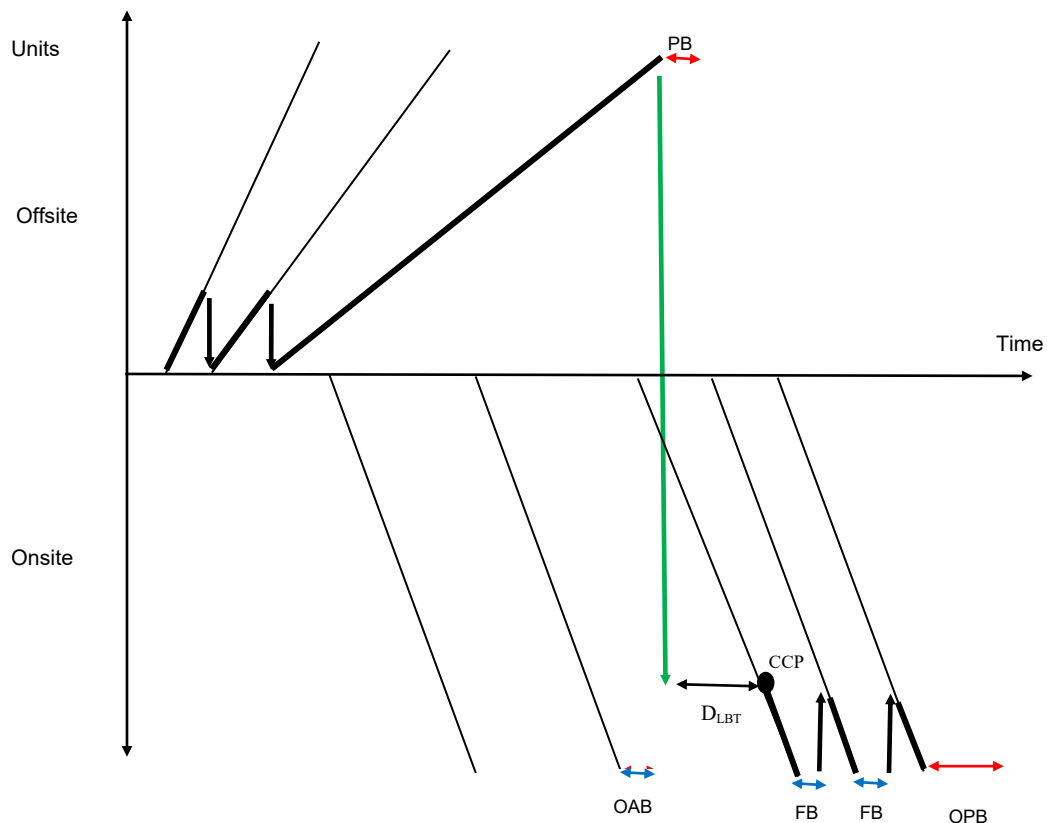


Figure 1: Integrated schedule for offsite and onsite activities.

3 DEVELOPED SOFTWARE

3.1 Design

Software design is explained in the next sections by introducing two levels of technical design. The first is high-level technical design, which discusses the main softwares utilized to build “Mod-Scheduler”. The

second is low-level technical design which discusses properties of inputs and sequential steps to generate the outputs drawn as the overall schedule for offsite and onsite activities.

3.1.1 High-level technical design

“Mod-Scheduler” was developed as a web application to benefit from tools and services brought by web technologies, which have become the norm in building software applications. The main advantage of web applications is their ability to be accessed from any computer connected to the internet, without needing to download or install developed software. Web applications are easily deployed and its versions are easily updated. They also benefit from third-party libraries, which draw interactive charts.

It was programmed as a three-tier application system comprising of: 1) web front-end as the presentation layer coded using Angular, JavaScript (JS), hypertext mark-up (HTML) and cascading style sheets (CSS) languages; 2) ASP.NET core back-end system written in C# to process the information, and 3) SQL server database to store the information.

The back-end system used the ASP.NET web application programming interface (API), which is a framework that facilitates building hypertext transfer protocol (HTTP) services to reach a broad range of clients, including browsers and mobile devices. “Mod-Scheduler” is currently self-hosted (works without a server to run the application), but can later be hosted on the cloud where it can be accessed over the internet. The front-end application and the back-end system communicate together using messages coded in JavaScript object notation (JSON) data format over an HTTP connection.

The back-end system was structured using a layered architecture as shown in Figure 2, which contains:

- Controller layer: Receives the request coming from the web application, performs authentication and authorization, and passes it on to service layer;
- Service layer: Processes incoming data, performs calculations and in case of data needing to be stored in database, and passes it to the repository layer; and
- Repository layer: Responsible for centralizing database operations.

The repository layer communicates with the database through an object/relation mapping (ORM) framework which is also called entity framework that enables developers to work with relational data as domain-specific objects. Using the entity framework, developers issue queries, using the language integrated query (LINQ) framework, and retrieve and manipulate the data as strongly typed objects.

3.1.2 Low-level technical design

The first input screen contains basic inputs related to offsite schedule, as shown in Figure 3 and Table 1, such as process name (process contains several activities), activity name, quantities/units, dependencies between activities, and lag durations. The user has the choice of entering aggressive and safe durations, or entering aggressive and safe productivity rates to calculate safe and aggressive durations. After the user completes the first input screen basic inputs, the second input screen is launched to enter the details of the onsite schedule. Toggling between different screens is easy using selected tabs at the top of each window. The third input screen connects offsite and onsite schedules by clarifying the position of the critical controlling point (CCP) between both schedules using the finish to start relationship between the end of an activity in offsite schedule and the start of an activity in onsite schedule. The front-end gathers this information for all offsite and onsite activities and displays them, as shown in Figure 3.

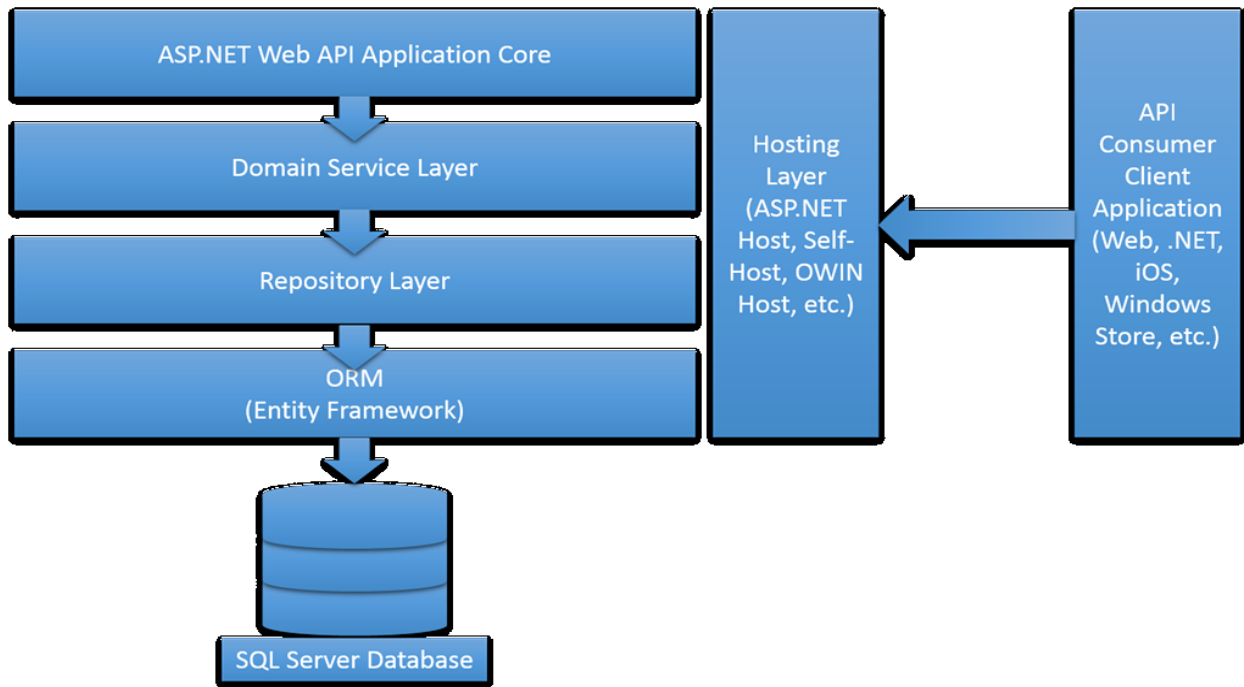


Figure 2: Architecture of high-level technical design

Mod Scheduler

Offsite Activities	OnSite Activities	Processes	Controlling Link	Graph						
<input type="button" value="Add Activity"/> <input type="button" value="Update Activity"/> <input type="button" value="Delete Activity"/>										
Name	Quantity	Duration	Agg Duration	Safe PR	Agg PR	Finish/Start	Quantity Delta	Function	Dependencies	Process
A1	15	15	7.5	1	2	0	0	1		Roof
A2	10	5	3.33333325	2	3	0	0	1	A1	Roof
A3	15	10	5	1.5	3	0	0	1	A2	Roof
A4	15	15	6	1	2.5	0	0	1	A3	Roof
B1	10	10	5	1	2	1	-15	1	A1	Wall
B2	10	7	2.33	1.42857146	4.291846	0	0	1	B1	Wall
B3	10	7	1.33	1.42857146	7.518797	0	0	1	B2	Wall
B4	10	10	8	1	1.25	0	0	1	B3	Wall
C1	10	10	5	1	2	0	-10	1	B1	COLUMN
C2	10	10	5	1	2	0	0	1	C1	COLUMN
C3	10	10	5	1	2	0	0	1	C2	COLUMN
C4	10	10	5	1	2	0	0	1	C3	COLUMN

Figure 3: Properties of activities

Table 1: Input properties

Name	Type	Description
Id	int	Technical identifier for activity
Name	string	For example, adding base columns
Units	float	Amount of work needed to complete the activity (quantities)
SafeDuration	float	Duration estimate + safety time
SafeProductivityRate	float	Set directly or computed based on Units divided by safe duration.
AggressiveDuration	float	Average duration estimate without safety time
AggressiveProductivityRate	float	Set directly or computed based on Units divided by aggressive duration
Process	Process	Links an activity to a process (example: walls, floor, ceiling)
StartToFinish	Float	Adds a time lag between the start of current activity and the end time of preceding activity.
Lag / Unit Delta	Float	Adds a unit lag between the start unit of current activity and the end unit of preceding activity.
Dependencies	List<Activity>	List of activities that depend on current activity. This determines relationship among activities.
Type	Int	Whether it is an offsite or onsite activity

3.2 Inputs

Interface of “Mod- Scheduler” includes four input screens for offsite activities, onsite activities, processes, and controlling link between offsite and onsite schedules as shown in Figure 3. It also includes one output screen named “graph” which generates the integrated overall schedule as shown in Figure 4. Inputs of the first screen for offsite activities are shown in Figure 5. Adding an activity to this screen requires ten inputs to be entered to the screen of offsite activities as follows: 1) name of activity, 2) quantity of this activity (e.g. 10 cubic meters of concrete), 3) the choice between inputting aggressive and safe productivity rates to calculate activity durations or inputting durations directly, 4) safe productivity rate or safe activity duration, 5) aggressive productivity rate or aggressive activity duration, 6) lag duration included for the finish to start relationship of this activity and preceding activity, 7) entering the Y axis coordinates of the start duration for any activity which is related to the Y axis coordinates of the finish duration of its preceding activity (e.g. if activity B1 is successor to activity A1 with a finish to start relationship, and quantity of activity A1 includes 15 units which means that the Y axis coordinates of its finish duration is 15 units , then a “quantity delta” of (-15 units) is entered so that the start of activity B1 starts at 0 for its Y axis coordinates), 8) name of process for each activity, 9) input of a duration function which can be either (max or none) to sequence activities that have two predecessor activities (e.g. if activity B2 has two predecessor activities A1 and B1 which are both related to activity B2 with a finish to start relationship, then the duration function is selected to be (max) so that the start of activity B2 equals the maximum of the finish duration of activities A1 and B1, and in case activity B2 has only one predecessor activity, then the duration function is selected to be (none)), 10) entering the name of activity that precedes the current activity.

Inputs for the second screen of onsite activities are similar to the required inputs for offsite activities, and this screen requires ten inputs to add an activity as the previously explained inputs for offsite activities. Names of work processes for offsite and onsite activities (e.g. floor, columns, etc.) are also inputs in the third screen. Inputs of the fourth screen for the controlling link between offsite and onsite schedules requires three inputs as follows: 1) name of offsite activity that connects to onsite schedule, 2) name of onsite activity that connects to offsite schedule, 3) transportation duration to transport panels/modules from offsite manufacturing facility to construction site. The output screen named “graph” draws both

offsite and onsite schedules before identifying the overall critical sequence or adding buffers as shown in Figure 6. Then by pressing the button named “add critical path” on this screen, the overall critical sequence is identified and the feeding and project buffers are identified automatically. The final output screen is shown in Figure 4.

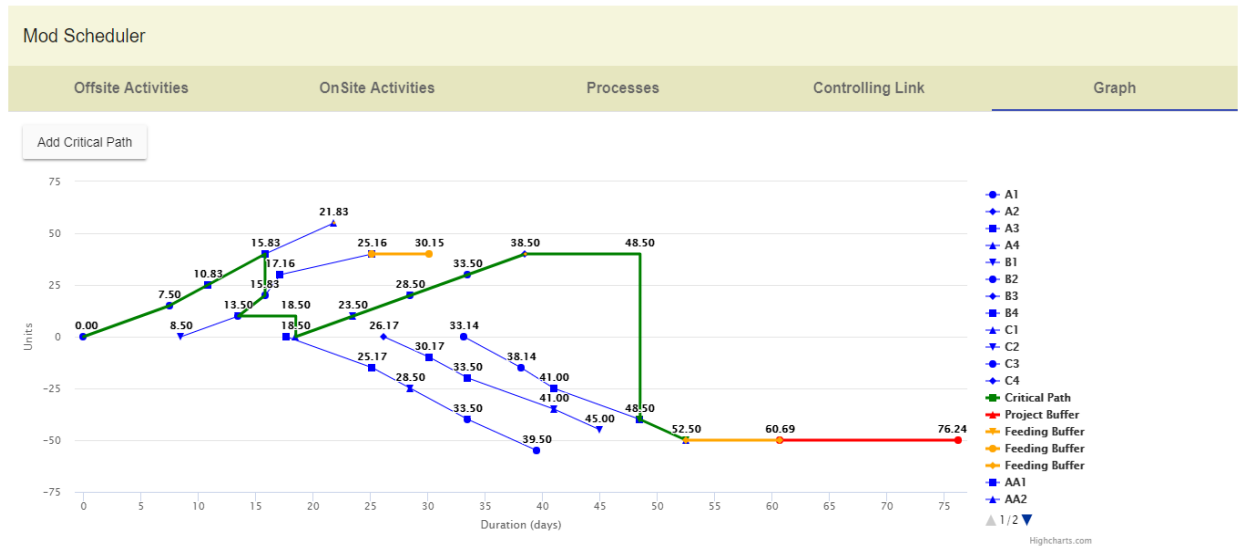


Figure 4: Overall schedule (software output)

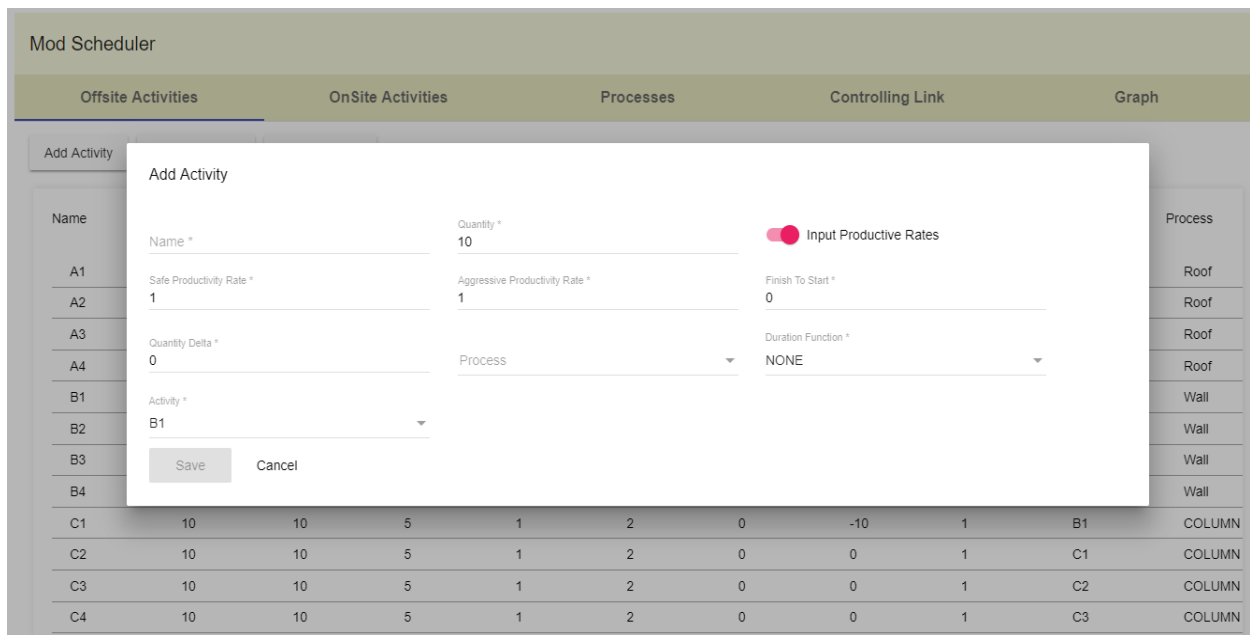


Figure 5: Inputs for offsite activities

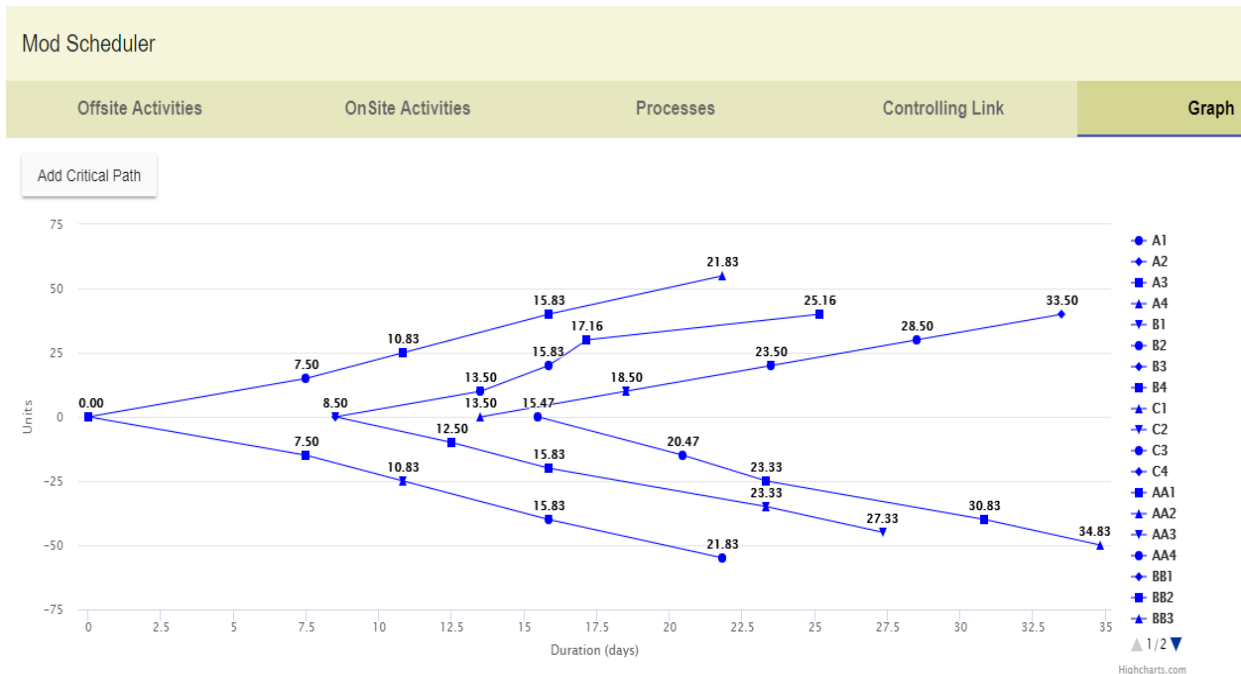


Figure 6 : Integrated schedule before adding buffers

“Mod-Scheduler” draws the overall schedule using inputs of activities by applying a set of equations and procedures at different levels according to the following five sequential steps:

- Step 1: Calculation of aggressive and safe activity durations for offsite and onsite activities

The overall schedule shows two schedules, an upper one for offsite activities and a lower one for onsite activities. Duration of each activity in both schedules is entered directly, or can be calculated using quantities of each activity and aggressive and safe productivity rates.

- Step 2: Sequencing of activities based on aggressive durations

The aggressive schedule is sequenced for each of offsite and onsite schedules separately. Then, the start of any activity is shifted if needed using time or unit lags through the X or Y axis respectively for the developed schedule.

- Step 3: Integration of offsite and onsite schedules

The critical controlling link between offsite and onsite schedules is determined by entering a finish to start relationship between an activity in offsite schedule and an activity in onsite schedule.

- Step 4: Identification of overall critical sequence

The procedure illustrated in (Salama et al., 2017b) to calculate multiple critical sequences for repetitive scheduling is coded for each of the offsite and onsite schedules separately. After the two schedules are connected using the critical controlling link, “Mod-Scheduler” determines the overall critical sequence for the overall schedule that generates the longest project duration after adding feeding and project buffers.

- Step 5: Adding feeding and project buffers

Project and feeding buffers are calculated for the overall critical sequence, represented as yellow and red lines, respectively. Activities and overall critical sequence are represented as blue and green lines, respectively, as shown in Figure 4.

4 Case Example

Three processes were assumed in this case example for the offsite schedule (sheathing, insulation, etc.), and another three processes were assumed for the onsite schedule (lifting, surveying, etc.) as shown in Table 2. Each process was conducted for four different sections of the modular building. Quantities of each section and work durations with confidence level of 50 and 90 percentile (D 50% and D90%) were the inputs needed to run the software as shown in Table 2. It is assumed that the transportation duration for modules is 10 days after finishing the fourth section of the third process in the offsite schedule (upper schedule on “Mod-Scheduler”), and that the controlling link should connect the third process of the offsite and onsite schedules. After loading the inputs into “Mod-Scheduler”, the integrated critical sequence between the offsite and onsite schedules is identified (shown as green line in Figure 4), and the project buffer is drawn automatically with a total duration of 15.55 days (shown as red line in Figure 4). Two feeding buffers are drawn with total durations of 5 and 8.19 days at the offsite and onsite schedules respectively. The total project duration is calculated to be 76.24 days after adding the feeding and project buffers as shown in Figure 4. The same inputs were used to validate these numbers using the method and equations provided by Salama et al. (2017a) on an excel sheet, and the same numbers for project duration and buffers were calculated.

Table 2: Assumed durations and quantities

Offsite	sections	First Process			Second Process			Third Process		
		Units (Quantity)	D (90%)	D (50%)	Units (Quantity)	D (90%)	D (50%)	Units (Quantity)	D (90%)	D (50%)
Offsite	1	20	12	7.5	15	10	5	15	11	5
	2	5	8.5	3.33	9	6	2.33	9	11	5
	3	15	9	5	6	4	1.33	6	11	5
	4	15	11	6	6	14	8	6	11	5
Onsite	1	20	12	7	15	9	4	20	10	5
	2	5	8	3.33	9	6	3.33	5	7.5	2.86
	3	15	9	5	9	13	7.5	15	12	7.5
	4	15	10	6	6	9	4	10	10	4

5 Conclusion

Scheduling software named “Mod-scheduler” is programmed specifically for modular and offsite construction using ASP.NET system coded with C# programming language. This software integrates offsite and onsite schedules while considering uncertainty of productivity rates for both schedules. The results of running this software were validated using an excel sheet calculations and proven to work effectively. Future work may include tracking and control automated method for the schedule generated from “Mod-scheduler”.

6 References

- Abu Hammad, A.A. 2003. A Decision Support System for Manufactured Housing Production Process Planning and Facility Layout. PhD thesis, Department of Civil and Environmental Engineering, the division of Research and Advanced Studies of the University of Cincinnati.
- Ajweh, Z. A. 2014. Framework for Design of Panelized Wood Framing Prefabrication Utilizing Multi-panels and Crew Balancing., Master of Science Thesis, Department of Civil and Environmental Engineering, University of Alberta.

- Altaf, M., Yu, H., and Al-Hussein, M. 2015. Online Simulation Modeling of Prefabricated Wall Panel Production using RFID System. Proceedings of the 2015 Winter Simulation Conference, Pages 3379-3390.
- Arditi, D., and Albulak, M. 1986. Line of Balance Scheduling in Pavement Construction. *J.Constr.Eng.Manage.*, 112(3), 411-424.
- Bu Hamdan, S., Barkokebas, B., Manrique, J.D., and Al-Hussein, M. 2015. A BIM-Based Simulation Model for Inventory Management in Panelized Construction. Proceedings, International Symposium on Automation and Robotics in Construction and Mining (ISARC), Oulu, Finland, Jun. 15-18.
- Francis, A., Miresco, E. T. 2013. Applying the Chorographical Approach to the Modelling of Multi-storey Building Projects." 4th CSCE Construction Specialty Conference, Montreal, Quebec.
- Harris, R. and Ioannou, P. 1998. Scheduling Projects with Repeating Activities. *J. Constr. Eng. Manage.*, 269-278.
- Hajjar, D., and AbouRizk, S. 2002. Unified Modelling Methodology for Construction Simulation. *Journal of Construction Engineering and Management*, 128(2), 174- 185.
- Hegazy, T., Moselhi, O., Fazio, P. 1993. BAL: An Algorithm for Scheduling and Control of Linear Projects. *Transactions of AACE International*, pg. C.8.1.
- Ioannou, P. G., and V. Likhitrungsilp. 2005. Simulation of Multiple-Drift Tunnel Construction with Limited Resources, Proceedings of the 2005 Winter Simulation Conference.
- Liu, H., Altaf, M. S., Lei, Z., Lu, M., and Al-Hussein, M. 2015. Automated Production Planning in Panelized Construction Enabled by Integrating Discrete-Event Simulation and BIM, ICSC15 – The Canadian Society for Civil Engineering's 5th International/11th Construction Specialty Conference.
- Liu, H., Lu, M., and Al-Hussein, M. 2016. Ontology-Based Semantic Approach for Construction-Oriented Quantity Take-Off From BIM Models in the Light-Frame Building Industry. *Advanced Engineering Informatics journal*, (30), 190–207.
- Lutz, J. D. 1990, Planning of Linear Construction Projects using Simulation and Line Of Balance, Ph.D. Thesis, Purdue University, West Lafayette, IN.
- Mansur, F. 1990. The Line of Balance Method." Major technical report, Center for Building Studies, Concordia University, Montreal, Canada.
- McGraw-Hill. 2011. Prefabrication and Modularization; Increasing Productivity in the Construction Industry. 54 pages, Massachusetts, USA.
- Modular Building Institute (MBI). 2016. Modular Advantage for the Commercial Modular Construction Industry. 59 pages, Charlottesville, Virginia.
- Moghadam, M., 2013. Lean-Mod: An Approach to Modular Construction Manufacturing Production Efficiency Improvement. PhD thesis, Department of Civil and Environmental Engineering, University of Alberta.
- Salama, T., Salah, A. and Moselhi, O. 2017a. Integration of offsite and onsite schedules in modular construction. Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC 2017), Taipei, Taiwan.
- Salama, T., Salah, A. and Moselhi, O. 2017b. Integration of Linear Scheduling Method and the Critical Chain Project Management. *Canadian Journal of Civil Engineering*, 10.1139/cjce-2017-0020.
- Srisuwanrat, C. 2009. The Sequence Step Algorithm: A Simulation-Based Scheduling Algorithm for Repetitive Projects with Probabilistic Activity Durations. PhD thesis in The University of Michigan.
- Taghaddos, H., AbouRizk, S., Mohamed, Y., and Hermann, R. 2009. Integrated Simulation-Based Scheduling for Module Assembly Yard. *Building a Sustainable Future*. pp. 1270-1279. doi: 10.1061/41020(339)129.
- Taghaddos, H., Hermann, U., AbouRizk, S., Y., Mohamed, A. 2012. Simulation-Based Multiagent Approach for Scheduling Modular Construction, *J. Comput. Civ. Eng.* 28 (2) 263–274.
- Xie, H., Shen, W., Hao, Q., Yu, H., Tang, Y. 2011. Production Simulation for Prefab Housing Facilities. 3rd International / 9th Construction Specialty Conference (CSCE), Ottawa, ON., pp. CN-189-1 – CN-189-10.
- Yang, I. 2002, Repetitive Project Planner Resource-Driven Scheduling for Repetitive Construction Projects, Ph.D. Dissertation, University of Michigan, Ann Arbor, MI.