The following publication Li, X., Shen, G. Q., Wu, P., Xue, F., Chi, H. L., & Li, C. Z. (2019). Developing a conceptual framework of smart work packaging for constraints management in prefabrication housing production. Advanced Engineering Informatics, 42, 100938 is available at https://doi.org/10.1016/j.aei.2019.100938.

Developing a Conceptual Framework of Smart Work Packaging for Constraints Management in Prefabrication Housing Production

Abstract

Constraints management is the process of satisfying bottlenecks to facilitate tasks assigned to crews being successfully executed. However, managing constraints is inherently challenging in prefabrication housing production (PHP), due to the fragmentation of processes and information during project delivery. Enlightened by the broadly accepted work packaging method and the smart construction objects (SCOs) model, this study aims to define and implement smart work packaging (SWP) for constraints management in PHP. Firstly, the framework of SWP-enabled constraints management (SWP-CM) with three primary functions, including constraints modeling, constraints optimization, and constraints monitoring, is established. In addition, this study develops a layered abstract model as a prototype representation to elaborate on the implementation of SWP for practitioners. Finally, a laboratory-based test is applied to validate the framework. It can prove that SWP indeed opens new avenues for smart constraints management for PHP.

Keywords: Smart Work Packaging (SWP), Prefabrication Housing Production (PHP), Constraints Management, Building Information Modeling (BIM)

1. Introduction

Prefabrication housing production (PHP) is an innovative approach that the prefabricated material, components, modules, and units are manufactured efficiently at different locations and then converge at the site for installation. This approach could alleviate the labor shortage and swiftly provide housings to mitigate the unbalanced housing supply and demand in Hong Kong (Li et al., 2018a; Wu et al., 2017). Although PHP has proven to be useful in the supply of public rental housing (PRH), it is still plagued by the pathological schedule delay which can lead to an adverse impact on Hong Kong's economic growth and competitiveness, particularly when manufacturing plants have been moved to the Great Bay Area of Mainland China. For example, the government planned to construct 13300 flat units of PRH in the financial year of 2016-2017. However, the actual amount of PRH production is 11276 units, and 15.22% delay occurred (Housing Authority, 2018). Dominant drivers for such delays have proven to be the uncertainties and constraints (Li et al., 2017a). Uncertainty means something that may occur, whereas constraint (e.g., limited space and buffers) is something that will happen. The constraints are the obvious bottlenecks and are more predictable than the uncertainties to be improved in task executions. Hence, reliable constraint-free workflows are vital for achieving an industrialized PHP environment across design, manufacturing, logistics, and on-site assembly so as to avoid schedule delays and cost overruns (Wang et al., 2016a).

The reliability of PHP schedules can be enhanced via proactive constraints management, which is the process of identifying, optimizing, and monitoring of bottlenecks to ensure that work package-level tasks assigned to crews can be timely and accurately executed. They can be related to technical sequencing, temporal/spatial limitations, and safety/quality concerns. Examples of such constraints include incomplete BIM models, drawings and specifications, unavailability of workforce, materials, prefabricated products, equipment and tools, shortage of temporary structures, limited workspace, lack of work permits, unidentified safety and hazard issues, uncontrolled environmental conditions (e.g., severe weather), untimely and inaccurate transportation, and uncompleted quality control. Managing constraints in PHP processes is to prepare more (e.g., on detailed and dynamic planning with lean solutions) and act fast (e.g., on decision-making) using available information and knowledge. As such, the primary objective of constraints management is to continually improve the reliability of workflow by guaranteeing that accurate information is always available at the right time in the right format to the right person. Currently, there have been numerous studies focusing on how to support decision makers and collaborative workers with precise and timely information for task execution (Zhong et al., 2017; Li et al., 2018b;). For instance, an internet of things (IoT)-enabled Building Information Modeling (BIM) platform is developed with the support of smart construction objects (SCOs) by equipping objects with information and communication technologies such as radio frequency identification (RFID), and by using augmented reality (AR), and other sensing and tracking technologies (Li et al., 2018c; Niu et al., 2016). Although Wang et al. (2016a) have made efforts to develop a framework by considering the adoption of information technologies for constraints management in oil and gas industry, there is so far no widely accepted approach for constraints management in PHP.

The development of smart work packaging (SWP) in recent years seems to be adequate to address the challenge. In PHP, there are a few studies which investigate the smart transformation of a group of tasks (e.g., the lowest level in the work breakdown structure) based on the building systems of product breakdown structure (PBS) by embedding the capabilities of visualizing, tracking, sensing, computing, networking, and reacting. The smart transformation centers upon autonomy, adaptivity, and sociability, which can facilitate better tasks execution by crews. For instance, the PHP machinery (e.g., cranes) can be augmented with the autonomy to transport or hoist the prefabricated products independently and without direct intervention from the surroundings (Chi et al., 2012). In addition, the PHP planning approaches can be enhanced with adaptivity to be capable of reacting resiliently through dynamic re-planning when constraints are not removed (Abuwarda and Hegazy, 2016). SWP can also be strengthened with sociability to interact in a peer-to-peer manner with other work packages or resources to collectively model the constraints (Taghaddos et al., 2012).

This study proposes and validates a new framework of SWP for constraint management in PHP based on the established theories of work packaging and SCOs (Isaac et al. 2017; Niu et al. 2016). Work packaging can break down the PHP processes into manageable pieces to facilitate execution of activities or tasks. SWP intends to improve the constraints during task executions in an autonomous, adaptive, and optimal manner. e.g., automatic identification and analysis of constraints and their interrelationships (Hamdi, 2013; Isaac et al. 2017), real-time sensing and tracking constraints status (Liu et al. 2015), and optimal constraints improvement planning in a dynamic manner (Abuwarda and Hegazy, 2016). SCOs are construction resources augmented with smart characteristics of awareness, communicativeness, and autonomy using emerging information technologies. However, SCOs are usually defined on single construction objects, without considering construction project operations such as work packaging. Thus, the development of SWP, as the integration of work packaging and SCOs, seems necessary and imperative to improve constraints management in PHP. To improve the shortcomings in current practices of constraints management, this study aims to develop a conceptual framework of SWP-enabled constraints management (SWP-CM) in PHP. The concrete objectives of this research are well explained below: (1) to define the SWP; (2) to establish the framework of SWP-CM; (3) to propose a functional structure of SWP as a layered system model; and (4) to validate the SWP-CM by a simulation game.

2. Background

2.1 Constraints Management

Constraints management (CM) is one of the critical strategies for production control and planning. The concept of constraint was firstly introduced in 1984 as the theory of constraints (TOC) which is a management philosophy for identifying the most critical bottleneck that prevents achieving a goal and then systematically improving the constraint until it is no longer the bottleneck (Goldratt and Cox, 1984). It assumes that each intricate system may comprise multi-connected activities, there is at least one activity that acts as a constraint in the fully connected system, and the entire process throughput can only be maximized when the constraint is improved. A corresponding deduction is that spending more time on optimizing non-constraints activities cannot generate significant benefits, and only improvements to the constraint will reach the goal. Thus, TOC aims to offer an accurate and continuous focus on improving the current constraint until it no longer confines the goal, at which point the focus moves to the next constraint. Constraints management systems have proven to be more effective when compared to the reorder-point systems and material requirements planning systems in the aspects of capacity management, inventory management, and process improvement in the manufacturing industry. It is also argued that constraints management can outperform the Just-in-time system owing to the more targeted nature of improvement efforts in constraints (Boyd and Gupta, 2004). However, there is still no sound approach to improve constraint management for achieving efficient collaborative working and decision-making at crew-level task executions. It is mainly due to the fragmented process and information in PHP, which may prevent the workers from agile constraints identification (Gong et al. 2019), adaptive constraints improvement (Abuwarda and Hegazy, 2016), and real-time constraints monitoring (Liu et al. 2015).

2.2 Work Packaging Method

TOC, to some extent, has similar philosophies as the Lean Construction. TOC uses its laser-like focus to improve the capacity, while Lean Construction uses the broad-spectrum tools to eliminate waste. In real practice, as PHP projects do not have infinite resources, an optimization process is needed to identify and improve the most critical constraints. In this instance, TOC can work as an efficient mechanism in prioritizing improvements for constraints, while Lean Construction can offer a rich toolbox of improvement techniques. Thus, the combination of TOC and Lean Construction may generate synergy on constraints management. The significance of integrating Lean Construction with constraints management to issue executable work plans has also been widely recognized by the construction industry. For example, work packaging is a planned, executable process to strategically decompose the PHP scope into distinct and manageable pieces with proper sizing and criteria. Each work package should be assigned to an individual supervisory unit that is able to handle all its constraints. Therefore, the tasks should be separated into smaller pieces (e.g., 500-2000 man-hours of work) so as the benefits outweigh the additional administrative burden (Isaac et al., 2017). Additionally, the most frequently used criteria in work packaging design include the type of prefabricated product, the workface in which the prefabricated product is located, the specific physical location of the prefabricated product, and the workflows (Ibrahim et al., 2009). The dependencies between tasks/activities included in various work packages should also be considered. Whereas the PHP can be broken down into a group of building systems (e.g., structure, envelope, partitions, services, and equipment) with a hierarchical product structure (e.g., material, component, module, unit) in the design, the work packaging in PHP can be defined by considering both product breakdown structure (PBS) and work breakdown structure. One of the practical examples is advanced work packaging (AWP), which was developed through the collaboration between the construction owners association of Alberta (COAA) and the Construction Industry Institute (Hamdi, 2013). AWP uses a hierarchy of engineering work packages (EWPs), construction work packages (CWPs) and installation work packages (IWPs) to allow engineering and procurement planning to be driven by construction sequencing. It breaks down the project processes into CWPs aligned with WBS. CWPs, in turn, contain one or more IWPs. However, the direct implementation of AWP in PHP may be limited. It works well in handling the complex mega project (e.g., oil and gas project), but its organizational structure with CWP, EWP, and IWP is hierarchical and not flattened enough for PHP to improve the efficiency of decision making and collaborative working (Li et al., 2019). Moreover, there are also several significant limitations in the current work packaging methods for efficiently managing constraints in PHP. Firstly, the process for identification and analysis of constraints and their interrelationships is sluggish because the constraints are only discussed in look-ahead meetings rather than in real-time manner (Hamdi, 2013; Isaac et al. 2017). In addition, constraints status is untraceable and non-transparent due to the lack of sensing and tracking technologies for monitoring (Liu et al. 2015). Constraints improvement planning is usually static without the dynamic replanning ability (Abuwarda and Hegazy, 2016). Enlightened by the smartness of smart construction object (SCO) (Niu et al., 2016), a more collaborative, autonomous, and adaptive approach for constraints management through constraints modeling, monitoring, and optimization may be possible.

2.3 Development of the Smart Work Packaging Method

Previous studies have made efforts to improve the smartness in the process management of prefabricated construction. For instance, Wang et al. (2016a) developed a framework for total constraints management in the oil and gas industry. However, information technologies were only conceptually discussed in their framework, and there was no validation (e.g., a prototype system) to demonstrate the smartness of the framework in constraints management implementation. In addition, Li et al. (2018a) investigated the stakeholder-associated risks to improve the reliability of phase-level scheduling. However, this study did not investigate constraints in the task-level plan, which are more predictable than the risks at the phase level. The on-site assembly service, developed by Li et al. (2018b), provided one of the services in the IoT-enabled BIM platform, which is a critical part to support smart work packaging (SWP). However, the platform cannot further divide the on-site assembly service into collaborative and manageable processes, therefore providing relevant work packages in each of the processes. Li et al. (2017a) developed a simulation game to test the learning effect of adopting information technologies and lean principles in prefabrication housing production process. Based on Li et al. (2017a), this study tries to enhance the work packaging method and constraints management in this simulation game to validate the proposed conceptual framework.

Much effort has also been made in using cutting-edge information technologies to make work packages smart (Ibrahim et al., 2009; Abuwarda and Hegazy, 2016). For example, Isaac et al. (2017) developed algorithms for BIM which can be integrated with design structure matrix and domain mapping matrix to automatically label relationships between prefabricated products and their following sequence in which the prefabricated products should be assembled. Table 1 demonstrates a summary of the studies related to the development of SWP. As shown in Table 1, the development of SWP has focused on the various aspects of constraints management, including modeling, monitoring, and optimization. Some studies, although not directly using the name "smart work packaging" or SWP, address the interaction between humans, resources, and the environment with smartness using emerging technologies such as IoTs, wireless sensor networks, big data, cloud computing, or other enabling technology to facilitate task execution.

<Insert Table 1 here>

Compared with traditional task execution process, SWP has many unique characteristics, including traceability, value-added, and awareness. However, information communication, adaptive to changes, autonomous actions during task executions have been identified as the necessary requirements of SWP in previous studies (Lu et al.2017; Wang et al. 2016b; Ren et al. 2017; Lee et al., 2009). Based on using simulated or historical data, SWP could achieve autonomy by executing particular tasks when specific requirements are met (Lu et al., 2017). In addition, each smart work package can gain sociability by communicating with its internal elements, as well as other smart work packages (SWPs) to work as a distributed multi-agent system for collaborative working (Ren et al., 2017). Most importantly, SWP must be adaptive and can react flexibly to changes by learning from its own experiences, environment, and interaction with others (Wang et al. 2016b; Lee et al., 2009). Thus, it is believed that the three critical characteristics of SWP are autonomy, adaptivity, and sociability. The potential functions of SWP have also been introduced and assessed in different scenarios including modeling (e.g., the understanding of the interconnections among tasks), monitoring (i.e. the tracking and updating of real-time status), and optimization (i.e. the planning and scheduling of tasks) (Luo et al. 2018; Wan et al. 2018; Zhang et al. 2018).

However, it should be noted that SWP and its definition, characteristics, functions , applications, and prospects in the PHP field have not yet been systematically explored for constraints management. Although individual SWP studies have been investigated, they do not provide a systematic view to explore the full potential of SWP, which is a necessity in driving toward a sweeping and interconnected smartness in next-generation PHP practice, particularly in the field of constraints management in PHP. This requires an investigation of the unique and inherent characteristics of SWP from the manufacturing industry and the incorporation of PHP characteristics.

3. Definition of SWP

In this study, SWP is defined as an approach to decompose the PHP workflows (e.g., technical process) by product breakdown structure (PBS) of building systems, and integrate *smartness* capabilities, such as visualizing, tracking, sensing, processing, networking, and reasoning into the workflows so that they can be executed autonomously, adapt to changes in their physical context, and interact with the surroundings to enable more resilient process.

The core characteristics of SWP, namely, adaptivity, sociability, and autonomy. Physical or functional information, such as shape, dimension, products type, the layout of the work section, work procedure, and positions of aids and resources, are not included because such information is also required in traditional work packaging method.

Adaptivity, the most distinct feature of SWP compared with traditional PHP work packaging method, denotes SWP’s ability to have a positive response to change, and learn from their own experiences, environment, and interactions with others. This characteristic is based on the concepts of smart workflows proposed by Wieland et al. (2008), which includes three dimensions, e.g., robustness, flexibility, and resilience (Husdal, 2010). Robustness is the fundamental feature level that the SWP can process. With robustness, SWP can quickly regain stability by accepting goal-directed initiatives when encountering constraints. It can be mainly applied to plan and control primitive tasks, which refer to elemental motion with few steps or short durations. For instance, the crane operator with the help of SWP can regain stable reaching, grasping, picking up, moving, and eye travel in the lift operations when encountering static constraint such as obstacles. Flexibility enables SWP to react to the foreseeable changes in a pre-planned manner. It is beneficial for guarding tasks execution against threshold-breaking or exceeding a pre-programmed tolerance range, and the SWP in this context primarily involves composite tasks such as to measure, connect, navigate, select, align, record, and report. For example, SWP can help crane operators measure the distance and report the parallax error when other tower cranes are approaching. Resilience is a high-level adaptivity that facilitates SWP to survive unforeseeable changes (that have severe and enduring impacts) in a dynamic replanning manner. The SWP tasks in this context include operation-specific tasks such as assembly, examining workflow, buffer layout, equipment path planning, and monitoring. For example, when an emergency occurs, SWP with resilience can offer assembly guidance and perform the optimized working path planning by cross-validating the real-time progress with as-planned workflow. Presently, SWP adaptivity can be achieved by advanced optimization approaches when making full use of the information collected from the sensing and tracking technologies.

Sociability ensures that SWP can communicate with the surroundings (e.g., other smart work packages (SWPs), human/machine/products in SWPs). The communication can happen at pull, push, or mixed modes. The pull mode occurs upon demand. For instance, the deliverables/information, such as prefabricated products from the transportation driver, are provided when requested by the SWP of the expeditor. In the push mode, SWP actively tracks and updates the information and issues alerts at regular intervals or when an emergency occurs. For example, the project manager of the SWP can obtain the traceability and visibility of the prefabricated products in a real-time manner to ensure its Just-in-time delivery. The mixed mode combines the pull and push to request and deliver information in a peer-to-peer manner. Apart from the three interaction modes of SWP, there are four relationships between SWPs, namely, composition, interface realization, inheritance, and dependency, which can enhance the sociability of SWP in handling the modular products/processes in PHP (Ramaji et al., 2016). Composition refers to the relationship of one SWP and its relevant SWPs. For instance, the work package of schedule management usually includes planning, progress checking, monitoring, and risk control. Interface realization refers to a group of work packages which support or rely on the behavior that is defined in an interface. Inheritance exists between a parent smart work package and its succeeding sub-SWPs. Dependency is the most popular relationship where the downstream SWPs are dependent on the upstream SWPs. To achieve the sociability of SWP, there are many communication and networking technologies to enhance the awareness of SWP such as active/passive RFID, ultrawideband (UWB), ZigBee, electromagnetic, Bluetooth, ultrasound, infrared (IR) proximity, Wi-Fi, near-field communication (NFC), laser, conventional radio frequency (RF) timing, wireless local area network (WLAN), received signal strength (RSS), and assisted GPS (A-GPS) (Niu et al. 2016; Zhang and Hammad, 2011).

Autonomy proposed in this study is based on the concept of SCOs (Niu et al., 2016). It refers to the capability of intelligent resources (e.g., machinery/tools/devices) in SWP to achieve autonomy through a pre-programmed method of decision making. There are three types of autonomy, including proactive autonomy, passive autonomy, and a mixed mode. Proactive autonomy aims to act in advance of a future situation. For instance, the autonomous crane tower can generate a lift plan in accordance with the dynamic construction environment. It can sense and monitor the dynamic constraints in the environment to predict and execute the plan in advance, without human interventions. Passive autonomy, on the other hand, can only perform instant reaction by a triggering mechanism, particularly triggered by the emergent situation due to the delays of personnel reactions. For example, the anti-heat stress uniform encapsulated in the SWP can issue an alert to the workers and help to reduce heat and humidity when they exceed a certain threshold (Yi et al., 2016). The mixed mode of autonomy may execute complex tasks involving multi-autonomy stages that can both control activities without intervention and act in a preset manner. For instance, the path planning in SWP of a crane operator can firstly be pre-programmed with optimal paths and collisions can be detected in the operation process with the dynamic autonomy.

The three core characteristics of SWP are interrelated. Each subclass of the adaptivity, sociability, and autonomy is not a bijection. Instead, various subclasses of characteristics can be integrated to address specific constraints. In more complicated scenarios, it is also possible that the integration of characteristics that are more advanced than these three features is needed. However, this is currently beyond the scope of this study.

4. Research Method

The development of the conceptual framework started with the definition of the SWP after a comprehensive review of the work packaging method, constraints management, and the smartness concept. Afterward, a draft paradigm, as shown in Figure 1, was proposed as the backbone of the framework. Constraint modeling is included in the SWP to facilitate the identification and interrelationship mapping of the constraints at the activity level (e.g., on-site assembly process). Then, the most influential constraint at the activity level to the goal (e.g., schedule performance) is isolated for further improvement, and this constraint often also contains many constraints at the task level (i.e., specific onsite operational activities). The constraints optimization service in SWP can help develop the optimal task executions by optimizing the constraints at the task level.

Tracking, updating, and predicting the statuses of the constraints at the task level are also included in the framework.

<Insert Figure 1 here>

In addition, a layered system model, as the functional structure of SWP in PHP, was also proposed to instantiate the conceptual framework. Its development is based on previous studies on IoT-enabled BIM platforms for PHP (e.g., Li et al. 2017a; Li et al. 2018b), in order to take advantage of both smart BIM platforms and smart construction objects in PHP.

Subsequently, the proposed framework and the layered system model were examined and finalized by 14 PHP industry professionals, who were the primary stakeholders of PHP in Hong Kong. All 14 experts investigated the framework and provided their comments on the potential application scenarios and functions based on their expertise. As shown in Table 2, the invited professionals included stakeholders from the client, contractor, manufacturer, transportation company, and consultancy. All industry professionals had more than 10 years of experience in the development, operation, and management of PHP projects and related technologies. It is therefore expected that these PHP professionals did provide an unbiased and constructive assessment of the framework.

<Insert Table 2 here>

In order to validate the proposed framework of SWP-CM, a laboratory test was also conducted by using a simulation game (named RBL-PHP, RFID/BIM/Lean-PHP, a role-playing game) developed by the authors (Li et al., 2017a). The following questions were raised:

- Can the constraints in PHP workflow be intelligently identified, improved, and monitored?
- Can the framework reduce project duration to improve the reliability of PHP workflow?
- Can productivity be increased in the implementation of this framework?

The aim of the game was to simulate a real-world PHP environment by building Lego^{TM} houses. The task goals were to construct four buildings with the shortest duration, the highest accuracy, and the maximum percentage of the plan complete (PPC). Figure 2 shows the roles and the number of people needed in this simulation game. All the 32 volunteers were postgraduate students with limited knowledge of SWP and constraints management, and ten of them had more than three years of working experience in the construction industry. Such an arrangement can help collect comments, suggestions, and insights from the perspectives of both academic scholars and industry practitioners. The volunteers were divided into two groups, who played in two separate rounds. The first round was related to the use of traditional planning and control (without SWP techniques), and the second round was related to the implementation of SWP-CM. These two rounds were then comparatively analyzed to demonstrate the benefits and differences in implementing the proposed framework. In order to reduce the influence by learning curve issues, there was a briefing session for both rounds, and the participants were also instructed to play before the game.

<Insert Figure 2 here>

5. The Framework of SWP-enabled Constraints Management (SWP-CM)

This section outlines the framework of SWP-CM, which aims to improve the workflow of PHP. After the review from selected industry experts, the client of HK Housing Society with the background of Lean Construction agreed that there are two levels of constraints in the PHP process, namely activity-level and task-level constraints, but he also pointed out that the framework should not only reflect the concurrent and continuous improvements of constraints from a perspective of Lean principles but also clarifies the process to the goal by identifying the critical chain of the constraints based on the theory of constraints. An expert from the contractor emphasized the alignment of work packaging stage among activity-level planning, task-level planning, and task executions. In addition, the expert from CIC highlighted the implementation of the three constraints management steps in this framework could help analyze the constraints and their interrelationships systematically in the whole activity process, along with providing the executable plan to remove the constraint at a more detailed level. However, the three steps of the framework should be well-defined in SWP. The IT consultancy mentioned the capabilities of IoT and emerging technology solutions and the integration of these technologies into the framework. A project manager from the client emphasized that the fusion of SWP and constraints management under a clear application scenario should be well considered.

Figure 3 presents the final version of the SWP-CM framework. The work packaging method with lean principles is designed as the basis to outline the workflow of the activity or task execution in PHP. In addition, the framework shows the three core modules of constraints management, followed by the detailed process of SWP-CM.

<Insert Figure 3 here>

To achieve the successful implementation of this framework, three functions, including constraints modeling, constraints optimization, and constraints monitoring must be well combined with the core characteristics of SWP for constraints management in PHP.

5.1 Constraints Modeling

Constraint modeling is a critical function with the sociability to allow a thorough understanding of interconnections among tasks or activities. There are three steps within this function. The first step is the constraints identification. The traditional process for constraints identification is static and usually executed once. The SWP can enhance this step in a passive autonomy manner by pre-programming the list of constraints and their classification with an open-data integration approach for constraints instantiation. Although each PHP project is unique, they share some similar types of constraints at the operational level (Li et al., 2018a), and it is possible to develop a database for organizing the potentially significant amount of constraints. Table 3 demonstrates the one example of constraints classification in the PHP process, which was sourced from the literature review and the on-site survey. These constraints are classified into manufacturing, logistics, and site constraints. Constraints such as incomplete design drawings/BIM models, approvals, and specifications are manufacturing constraints, which restrict the subsequent activities in logistics and on-site assembly. Logistics constraints contain limited weight and height for vehicles on the road, unavailable production schedule, and transportation schedule. Without JIT deliveries, the site buffer may be congested, or underutilized and on-site assembly cannot be efficiently executed. Site constraints include inadequate buffer and workspace, unavailable and unassigned labor resources, lack of collision-free crane path planning, lack of optimal installation sequence, and adverse weather conditions. The reason for this classification is that Manufacturing, logistics, and on-site assembly are the most critical stages in PHP, which can facilitate crews to identify the constraints in their stages. Once the list is embedded into the SWP, a set of pre-defined constraints and their relationships will be available for critical constraints identification.

<Insert Table 3 here>

The second step is the constraints relationship mapping. In real PHP projects, constraints are usually not independent and may have dynamic interrelationships. As such, a thorough understanding of these relationships is necessary. Figure 4 shows a simple example that includes only one crew with SWP in each selected trade (e.g., manufacturing worker, transportation driver, expeditor, buffer foreman, crane operator, installation worker). The constraints for production (e.g., drawings, BIM models, specifications, machinery) can be handled in the SWP of manufacturing worker. The development of SWP for expeditor needs to rely on well-satisfied constraints of vehicle locations, production, and transportation schedule in SWP of transportation driver. Therefore, any failure of constraints improvement in each SWP may lead to subsequent SWP delay in task executions. The control theory-based system dynamics (SD) model have the capacity to analyze the interactions (e.g., casual loop) and structures (e.g., stock and flow) of the project environments due to their perfect representation of feedback effects. SD models are primarily linked to strategic level context, such as the satisfaction level of the tasks, level of worker fatigue, level of worker skill. The Discrete Event Simulation (DES) can simulate sequential operation details and offer detailed information for execution. Taking the on-site assembly process as an example, the DES model may include detailed information such as the capacity and number of project resources, the duration of on-site assembly tasks, and the lifting distance of the crane tower. Thus, the hybrid SD-DES model can be an alternative to be incorporated into SWP to facilitate the constraints relationship mapping. The last step is the constraints scenario analysis, which can be presented in the interface of SWP for both project managers and workers to show the different simulation results on the schedule performance by evaluating the influence of different critical constraints. The most influential one will be selected for further optimization and monitoring.

<Insert Figure 4 here>

5.2 Constraints Optimization

5.3 Constraints Monitoring

In PHP projects, the latest constraints information is essential for the superintendent or workers to check the progress and issue constraint-free SWPs. As such, real-time constraints monitoring is needed. There are three processes within the function of constraints monitoring. The first process is constraints tracking, which focuses on tracking each individual constraint. For tracking purposes, a mixed type of autonomy is preferred. For instance, the availability of prefabricated products can be tracked by both active and passive RFID (or IoT systems) and visualized in the BIM as the interface of SWP (Li et al., 2018b). The second process is constraints status updating, which concentrates on computing the maturity of a task. The maturity index can be used to support short-term decision-making in a mixed type of sociability. As shown in Fig.5, Fig.6, and Fig.7, it is the interface of a smart work package for the site expeditor. Firstly, it can enable site expeditor with the ability to update the status of the prefabricated products' locations in a real-time manner. Fig.5 shows each prefabricated product with their ID, status (produced, arrived, or erected), time, latitude, and longitude measured by GPS. At the same time, the digital twins (e.g., BIM models) of smart objects (e.g., prefabricated products mounted with RFID and GPS) can be visualized at regular intervals or via ad-hoc networking on the expeditor interface of SWP for monitoring (as shown in Fig.6). Additionally, it can display locations of trucks in the google map and reveals the task maturity of logistics associated smart work packages for each truck and driver by three status (truck loading, cross-border, arrived) in Fig.7. This can guarantee the prefabricated products being transported to achieve JIT delivery, i.e., the pull perspective. The final process within this function is constraints predicting and alerting. The constraints alerting aims to warn the variations by comparing as-planned constraints improvement plan and real-time constraints status. Historical variation can be used to train and predict the next variation in a robust manner.

<Insert Figure 5 here>

<Insert Figure 6 here>

<Insert Figure 7 here>

6. The Functional Structure of SWP: Layered System Model

To achieve the characteristics and functions of SWP, a three-layered system is proposed (See Fig.8).

<Insert Figure 8 here>

The *context provisioning layer* (CPL) is capable of managing the context information of PHP processes, which is often referred as both physical and functional information (e.g., dimension, quantity, specifications, location, resources status). For CPL, BIM platforms can be adopted because it has proven to be an effective digital platform to offer users with the ability to generate, integrate, analyze, simulate, visualize and manage the physical and functional information of a facility (Li et al., 2017b). In addition, it can also support the development of various context-aware applications through application programming interfaces (APIs). The BIM models can also be used to integrate context from multiple sources (e.g., dynamic sensor data, smart construction objects, internet of things) for value-added services. The BIM models can be utilized to break down the design into many units, and each unit comprises various materials, components, and modules. All the prefabricated products within a unit can be grouped into a product work package (PWP), which is in accordance with the product breakdown structure of building systems. The PWP will then be decomposed into SWPs by integrating the context of the workflows (process), work faces (location), duration, and resources.

The *context integration layer* (CIL) adopts the output of CPL to accommodate information, algorithms, and functions into more advanced representations and provide domain-specific functions needed by SWPs. Compared with CPL, there is no off-the-shelf system for CIL. The primary contribution of this model is to present the concept of how to design this layer. There are two context integration processes (CIPs) for CIL, namely (1) Core CIPs, and (2) Domain-specific

CIPs. Within a location-based workflow engine, the former can help map the physical products, data, and services into the specific location-based workface to integrate the necessary elements for work packages, while the latter can help workers with different domain knowledge extract well-formatted work packages from Core CIPs and access different functions. In the core CIPs, the BIM model can be decomposed into various prefabricated products with both physical and functional information, which can form different product work packages (PWPs). Then, these PWPs can be integrated into the workflow of the PHP process (e.g., on-site assembly process). At this moment, the process-oriented information, e.g., the location of workface, technical procedure, required resources, can be integrated with PWPs to generate the work packages by introducing advanced algorithms (e.g., partitioning algorithms). The integration of PWPs with workflow by CIPs serves an autonomous pattern. A Core CIP receives a call from the workflow (a higher-ranking Core CIP of upstream SWPs) and remodels the request to the required format of the service including context query, insert, manipulation, and event. Context queries facilitate the query to be synchronized with context information, e.g., with a query language. The query result can serve as a variable to be injected into the complex workflow. If the query language allows data manipulation, a workflow can enable the function of context insert and change. The second process is related to domain-specific CIPs and can offer context information at various semantical levels for SWPs. The domain-specific CIPs include two primary functions: one is to merge specific functional elements to the well-formatted work packages from core CIPs to form SWPs; the other is to simplify the interfaces (e.g., web service interface) of SWPs for accessing their functionality.

Finally, the SWP layer (SWPL) can not only issue a smart work package with mobile, wearable, and executable capacity but also provide a platform to interact with other SWPs. In addition, any execution failure can trigger the dynamic re-planning function to provide more adaptive SWP. The experts also evaluate the proposed layered system model by their expertise and project experience, and the comments are summarized as follows: “This functional structure of SWP fully utilizes the capabilities of existing BIM platforms and smart construction objects to help equip the workers with more value-added information and make them more skillful on task executions.” (senior IoTs engineer, TSL) “It is feasible to embed this layered system model into the service-oriented architecture of the previous project ‘IoT-enabled BIM Platforms for Prefabrication Housing Production.’” (senior BIM system architect, Gammon Construction)

7. Validation

7.1 Validation Design

A simulation game following the real processes of PHP projects (e.g., a Subsidized Sale Flats project owned by the Hong Kong Housing Society and locates at 48 Chui Ling Road, Tseung Kwan O Area 73A) is conducted through a workshop to assess the validity of the proposed framework. According to the role setting and the proposed framework of SWP-CM, 14 SWPs were developed for the simulation game (See Figure 2 and Table 4). There are three connected scenarios (manufacturing, logistics, and on-site assembly) in this game. A process map was provided to the participants to understand the simulation game. In this study, 13 constraints, including lack of approvals from site manager, design drawings, BIM models, specifications, tools, production schedule, transportation schedule, prefabricated products (e.g., material, components, modules, units), buffer space, assembly instructions, quality and inspection hold-points, crane lift and place location, and vehicle limitation in weight and height, were included. If the project team cannot improve these constraints in an efficient manner, the game may suffer delay.

<Insert Table 4 here>

The first round of the game focused on the SWP-CM framework. The constraints identification process was conducted to synchronize the constraints list and the constraint relationship map to the SWP, which could be accessed by each participant through mobile devices. This process was achieved at the beginning of the game in the social network analysis (SNA) service of SWP, which included three primary steps: (1) The participants registered in the SNA service of their own SWP and accessed the full list of constraints; (2) The participants scored and evaluated the constraints interrelationships; (3) The participants visualized the constraint network and identified critical constraints and constraint interactions. After the identification, a hybrid system dynamic (SD)-discrete event simulation (DES) model service was adopted to assess and simulate the potential effect of the identified constraints on the schedule performance. DES was adopted to measure the operation level of game and SD was related to the strategic level consideration, including resource availability, operation efficiency, and schedule performance. Finally, the constraints analysis results were also demonstrated to participants by embedding the results in specific SWP buttons. As shown in Figure 9, when clicking "Expeditor_SWP," the expeditor could find all related constraints and other interactional SWPs. After clicking the specific constraint in each SWP, the simulation results can be presented. Apart from the constraints modeling, the detailed task execution plans for improving each constraint are also presented. Lean principles, such as pull methods, Just in time delivery, and standardized work, served as the optimization strategies in this simulation game. For instance, the pull method can be used to improve the constraints "lack of production schedule" in the SWP_11 (See Figure 9) for expediting the production process. Furthermore, the status of each constraint was also tracked and visualized through the use of RFID tracking technology and BIM visualization interface (see 10.2 "prefabricated products traceability" in Figure 9). With SWP-CM implementation, Group A was able to detect and analyze all constraints in the first 9 minutes and adopt relevant optimization strategies. The first round took 35 minutes, and the performance of Group A was evaluated by the percentage of plan complete (PPC), productivity index, and extra cost. The definition of these three indicators and their calculations are shown in Table 5.

<Insert Figure 9 here>

<Insert Table 5 here>

The second round game focused on the traditional constraints improvement method. The following changes were made, while other conditions remained the same.

(1) Constraints modeling, including the relationship map and analysis results, were not provided to Group B. Based on the inputs of the 14 industry professionals, constraints identification, relationship mapping, and analysis were conducted informally on the basis of experience.

(2) Constraints optimization strategies were only developed when the constraints happened. The participants could discuss optimal solution strategies in a meeting when constraints occurred.

(3) The players were not allowed to directly monitor others who have geographical barriers in real situations. In this simulation, they can arrange regular coordination meetings to report their own progress.

As there was no implementation of SWP-CM, the 13 constraints had not been timely identified until the second 9-minute interval. The game suffered delay due to the late removal of the constraints (e.g., shortage of tools and prefabricated products) and the performance was also measured by the same indicators.

7.2 Validation Results

The results are shown in Tables 6-8, respectively. Table 6 demonstrates the actual duration and the PPC values of the two rounds. A total of 35 min was recorded in the first round while the second round took 45 min, which suggests that 22.2% reduction in project duration was achieved through the implementation of SWP-CM. The main underlying reason was the late identification and improvement of the constraints in the second round, and participants spent more time understanding the constraints and identifying optimization strategies. Table 7 shows the results of the simulation game at extra cost. An extra cost of $7460 was recorded in the second round while there was no extra cost in the first round. In the second round, as the push system without constraints monitoring was adopted, two additional units were produced, and one unit was manufacturing-in-process (MIP). Table 8 shows the productivity index of the two rounds. The productivity is significantly improved in all three phases, including manufacturing (P_m : 0.53 → 0.67; 26% increase), logistics (P_l : 0.88 → 1; 14% increase), and on-site assembly (P_a : 0.49 → 0.65; 33% increase). Efficient information sharing and communication in the first round demonstrated the effectiveness of the real-time constraints modeling, optimization, and monitoring, which can be considered as the main contribution to the increase in productivity.

<Insert Table 6-8 here>

In summary, the round with SWP-CM outperforms the traditional round. The results also answer the previously raised questions with the following evidence: (1) Several intelligent techniques (e.g., SNA, DES, SD, Lean tools, BIM) have been used in constraints modeling, optimizing, and monitoring to achieve the certain level of sociability, adaptivity, and autonomy in the SWP-CM round; (2) The duration was reduced by 22.2% in in the SWP-CM round and $7460 extra cost occurred in the traditional round; (3) The productivity in the phases of manufacturing, logistics, and on-site assembly was increased with 26%, 14%, and 33%, respectively.

8. Discussion

Constraints management in modern PHP projects is essential because PHP processes are separated into different stages. Existing approaches to constraints management have several shortcomings, including low transparency of constraints status, and non-optimal or inflexible constraints improvement planning (Wang et al. 2016a). The previous manual and people-centric approaches in constraints management disregard the potential of IT to accurately, timely, and agilely in managing constraints, thus enabling the reliable workflow in PHP scenarios. With *smart* characteristics, including adaptivity, sociability, and autonomy, SWP can strengthen constraints modeling, monitoring, and even optimization. Accordingly, SWP can improve human deficiencies or skills in tasks execution to save time and cost. SWP can identify and analyze the latest constraints in a pull or push manner, provide optimal constraints improvement planning at different levels such as robustness, flexibility, resilience, and track, update, and predict the constraints status autonomously.

SWP provides an immense opportunity to improve workflow management in the global modular/prefabricated construction industry. SWP can significantly enhance the power of object-oriented BIM, which has been broadly recognized as a potential of integrating physical objects of product-oriented PHP and informational components to form situation-integrated analytical systems which can respond intelligently to the dynamic changes of real-world scenarios and offer data-oriented lean solutions (Li et al. 2017b). Current BIM models are mostly created in an as-designed condition, with updates in the subsequent stages including construction and maintenance. To make BIM a handy information hub in tasks execution with data-oriented lean solutions, as-built information is urgently needed to timely exchange with BIM. Presently, as-built data updates are primarily based on manual site survey or fragmented information technologies adoptions, which are time-consuming, error-prone, and non-value added information (Shrestha and Behzadan, 2018). To some extent, BIM development for physical project execution has come to a bottleneck with as-built information being synchronizing between BIM and tasks execution in a real-time and value-added manner to support constraints management. SWP can be adopted to bridge the value-added information gap between BIM and information technologies supported objects (e.g., smart PHP objects). The sociability of SWP means that they can interact with other SWPs or synchronize as-built information with BIM in a pull or push manner, and the adaptivity of SWP can make them respond to changes in a robust, flexible and resilient manner. The characteristic of autonomy enables SWP to respond in a proactive or passive manner.

Given the capacity of SWP to interact with other platforms, SWP can also benefit from the development of the Internet of Things (IoT), an emerging paradigm that has attracted considerable attention in the lifecycle of PHP (Li et al., 2018b), In the IoT paradigm, the constraints status can be connected at any time and anywhere. The gateway, an IoT-enabled industrial computer, can provide a communication link between physical sensors and SWPs. Thus, IoT can enable the SWPs to be a loosely coupled, decentralized, multi-agent system. The adaptivity held by SWP is a core property in the IoT ecosystem, as the flexible and resilient actions can make the planning and control of constraints more dynamic. With the characteristic of autonomy, SWP can connect with and handle the autonomous objects (e.g., vehicle, crane, robotics, 3D printer) based on specific protocols, e.g., a fill-up based trigger (Wu et al., 2016). Once the smart workflow is established, information sensed by each autonomous object can be shared with SWP in a proactive manner. These all contribute to the underpinning philosophy of construction industry 4.0 (Longo et al., 2017).

Furthermore, a smart work package can be generated from BIM by decomposing the BIM models and integrating the functional information such as tasks sequence, workflow, resources, location with the decomposed physical information including building systems and prefabricated products. Its information can be pulled out from context provision layer for assisting constraints modeling (e.g., automatic analysis of the topological constraints and their interrelationships), optimization (e.g., visual guidance and interactive representation of the work sequence can be obtained by applying optimal lean solutions), and monitoring (e.g., the resource requests can be evaluated and monitored in a real-time manner). The functions of SWP are developed and integrated into the context integration layer in a specific format (e.g., ifcXML), which can be connected to BIM. Files using the IFC schema can be interoperated on BIM platforms, which facilitates better information sharing and exchange (Lee et al. 2016). SWP also reduces manual operations, including reformating or reinterpreting information (e.g., constraints status) when using BIM, thus eliminating the possibility of the error caused by human intervention during data processing. It is envisaged that the proposed SWP can address the bottleneck that limits BIM expansion and present opportunities to make BIM a genuinely dynamic workflow management system rather than the static model management system.

It can be envisaged that SWP will progressively override conventional PHP constraints management to develop into an effective workflow management approach in the future. However, there are still numerous challenges to face. Firstly, from an organizational perspective, there will probably be resistance to diverge from the current constraints management practices in order to embrace smartness. Meanwhile, although SWP can help simplify interface management between tasks/activities carried out by different sub-contractors, the adoption of SWP for constraints management is more challenging in PHP projects with multiple tiers of subcontractors. Secondly, from a technical perspective, the interoperability of SWP will also be a challenge. The smartness of SWP relies on efficient data exchange. Without a universal standard for SWPs, there will be no smartness (though presently SWP can be operated based on BIM interfaces which are interoperated through ifcXML). The PHP industry is also fragmented. No individual can drive the industry toward fully integrated advanced technologies development and adoptions (Niu et al., 2016). The third challenge, from an economic perspective, is the expense of developing and deploying SWP. The PHP industry is comparatively slow-moving to embrace the new wave in the adoption of new technologies, and organizations within the industry would be very sensitive to expand on new technologies.

9. Conclusion

PHP has fragmented processes, which may generate various constraints in the critical chain of PHP. If the constraints cannot be timely improved, the reliability of workflow may be affected, and schedule delay and cost overrun will occur. The primary contributions of this study to the body of knowledge are threefold. Firstly, Inspired by the theories of work packaging and SCOs, SWP is defined as PHP workflows which are decomposed in accordance with PBS of building systems that are made smart by augmenting with the capacities of visualizing, tracking, sensing, processing, networking, reasoning so that they can be executed autonomously, adapt to changes in their physical context, and interact with surroundings to enable more resilient process. Secondly, equipped with three characteristics sociability, adaptivity, and autonomy, a continuous improvement framework for constraints management with three functions, including constraints modeling, constraints optimization, and constraints monitoring is proposed and illustrated by several examples and scenarios.The rationale and methodology in the framework of SWP-CM can be generalized because the development of the framework does not rely on identifying and removing specific types of constraints.Thirdly, a formal structured SWP representation is proposed by developing a layered system model involving context provisioning layer (CPL), context integration layer (CIL), and smart work packaging layer (SWPL) to realize these three functions.

Results from the validation process signify the benefits when implementing the framework of SWP-CM in PHP. 22.2% reduction of project duration was achieved, and no defective units were generated in the round of SWP-CM. Productivity was also improved, particularly in the manufacturing and on-site assembly stage. Thus, it can be concluded that SWP provides enormous opportunities to improve constraints management in PHP, particularly in conjunction with BIM. It can extract the context information (both physical and functional information) of product work packages from CPL (BIM platforms integrating with IoT). It can also insert the value-added as-built information into the BIM platforms in a pull or push manner. SWP can also be combined with the IoT-enabled gateway to act as a loosely coupled, decentralized, multi-agent system to make the status of the constraints be connected at any time and anywhere.

However, It should be noted that SWP for constraints management is in the early stage of its development. There are several barriers to the development and implementation. For example, there are technical difficulties related to the integral approach in constraints identification and interrelationship mapping, the efficient algorithms for dynamic re-planning in constraints optimization, and robustness hardware (e.g., autonomous robots, vehicles, cranes) and software (location-based workflow engine, interoperability of connected system) for constraints monitoring. There are also challenges related to technology acceptance, organizational changes, and cost issue.

By overcoming these challenges, it is believed that SWP can help establish safer, more adaptive, more proactive, more efficient, and more sustainable PHP workflows.

Acknowledgments

This research was funded by the Australian Research Council Discovery Project (grant number No. DP180104026), the Linkage Project (grant number No. LP180100222) and the National Key R&D Program of China (No.2016YFC070200504). It was also supported by the Research Institute for Sustainable Urban Development of the Hong Kong Polytechnic University, National Natural Science Foundation of China (No. 71801159), and Natural Science Foundation of Guangdong Province (No. 2018A030310534).

Glossary

Product Breakdown Structure (PBS): It is a product-oriented planning approach to analyze, document and communicate the outcomes of a project, which offers a comprehensive understanding of the physical deliverables. (Highlights: showing the physical deliverables)

Work Breakdown Structure (WBS): It is a deliverable-oriented planning tool to hierarchically decompose the entire scope of work into the combination of product, data, and service that are required in a project. (Highlights: showing the work required to produce deliverables)

Advanced Work Package (CWP): It is a planned, executable process that encompasses the work on an EPC project, beginning with initial planning and continuing through detailed design and construction execution. (Highlights: showing the framework of construction execution)

Construction Work Package (CWP): It is an executable construction deliverable with the well-defined (e.g., budget and schedule) work scope which cannot overlap with another construction work package.

Engineering Work Package (EWP): It is an engineering deliverable with preparation-oriented work scope, which includes drawings, procurement deliverables, specifications, and vendor support to be consistent with the sequence and schedule of CWPs.

Installation Work Package (IWP): It is a detailed execution plan that ensures all necessary elements used to complete the scope of the IWP are well organized and delivered before executions to enable workers to perform quality work in a safe, effective and efficient manner.

Smart Work Packaging (SWP): It is defined as an approach to decompose the PHP workflows (e.g., technical process) by product breakdown structure (PBS) of building systems that are made smart with augmented capacities of visualizing, tracking, sensing, processing, networking, and reasoning so that they can be executed autonomously, adapt to changes in their physical context, and interact with the surroundings to enable more resilient process.

Reference

Abuwarda, Z., & Hegazy, T. (2016). Work-Package Planning and Schedule Optimization for Projects with Evolving Constraints. *Journal of Computing in Civil Engineering, 30*(6), 04016022.

Blanco-Novoa, O., Fernández-Caramés, T. M., Fraga-Lamas, P., & Vilar-Montesinos, M. A. (2018). A practical evaluation of commercial industrial augmented reality systems in an industry 4.0 shipyard. *IEEE Access, 6,* 8201-8218.

Boyd, L., & Gupta, M. (2004). Constraints management: what is the theory?. *International Journal of Operations & Production Management, 24*(4), 350-371.

Chi, H. L., Chen, Y. C., Kang, S. C., & Hsieh, S. H. (2012). Development of user interface for teleoperated cranes. *Advanced Engineering Informatics, 26*(3), 641-652.

Giner, P., Cetina, C., Lacuesta, R., & Palacios, G. (2012). Enabling Smart Workflows over Heterogeneous ID-SensingTechnologies. *Sensors, 12*(11), 14914-14936.

Gong, P., Teng, Y., Li, X., & Luo, L. (2019). Modeling Constraints for the On-Site Assembly Process of Prefabrication Housing Production: A Social Network Analysis. *Sustainability, 11*(5), 1387.

Goldratt, E. M., & Cox, J. (1984). The Goal, Croton-on-Hudson. *NY: North River Press Inc.*

Hamdi, O. (2013). *Advanced Work Packaging from project definition through site execution: driving successful implementation of workforce planning* (Doctoral dissertation, The University of Texas at Austin). Retrieved from https://repositories.lib.utexas.edu/handle/2152/21384

Housing Authority (2018), Number of Applications and Average Waiting Time for Public Rental Housing, https://www.housingauthority.gov.hk/en/about-us/publications-and-statistics/prh-applications-average-waiting-time/

Husdal, J. (2010). A conceptual framework for risk and vulnerability in virtual enterprise networks. In *Managing risk in virtual enterprise networks: implementing supply chain principles* (pp. 1-27). IGI Global.

Ibrahim, Y. M., Lukins, T. C., Zhang, X., Trucco, E., & Kaka, A. P. (2009). Towards automated progress assessment of work package components in construction projects using computer vision. *Advanced Engineering Informatics, 23*(1), 93-103.

Isaac, S., Curreli, M., & Stoliar, Y. (2017). Work packaging with BIM. *Automation in Construction, 83,* 121-133.

Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., & Ivanova, M. (2016). A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. *International Journal of Production Research, 54*(2), 386-402.

Lee, K., Paton, N. W., Sakellariou, R., Deelman, E., Fernandes, A. A., & Mehta, G. (2009). Adaptive workflow processing and execution in Pegasus. *Concurrency and Computation: Practice and Experience, 21*(16), 1965-1981.

Lee, Y. C., Eastman, C. M., & Solihin, W. (2016). An ontology-based approach for developing data exchange requirements and model views of building information modeling. *Advanced Engineering Informatics, 30*(3), 354-367.

Li, C. Z., Xu, X., Shen, G. Q., Fan, C.,Li, X., Hong, J. (2018a). A model for simulating schedule risks in prefabrication housing production: a case study of six-day cycle assembly activities in Hong Kong. *Journal of Cleaner Production, 165*, 10

Li, C. Z., Xue, F., Li, X., Hong, J., & Shen, G. Q. (2018b). An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction. *Automation in Construction, 89,* 146-161.48-1062.

Li, X., Shen, G. Q., Wu, P., Fan, H., Wu, H., & Teng, Y. (2017a). RBL-PHP: Simulation of Lean Construction and Information Technologies for Prefabrication Housing Production. *Journal of Management in Engineering, 34*(2), 04017053.

Li, X., Wu, P., Shen, G. Q., Wang, X., & Teng, Y. (2017b). Mapping the knowledge domains of Building Information Modeling (BIM): A bibliometric approach. *Automation in Construction, 84,* 195-206.

Li, X., Yi, W., Chi, H. L., Wang, X., & Chan, A. P. (2018c). A critical review of virtual and augmented reality (VR/AR) applications in construction safety. *Automation in Construction, 86,* 150-162.

Li, X., Shen, G. Q., Wu, P., & Yue, T. (2019). Integrating building information modeling and prefabrication housing production. *Automation in Construction, 100,* 46-60.

Liu, H., Al-Hussein, M., & Lu, M. (2015). BIM-based integrated approach for detailed construction scheduling under resource constraints. *Automation in Construction, 53,* 29-43.

Longo, F., Nicoletti, L., & Padovano, A. (2017). Smart operators in industry 4.0: A human-centred approach to enhance operators' capabilities and competencies within the new smart factory context. *Computers & Industrial Engineering, 113,* 144-159.

Lu, S., Xu, C., Zhong, R. Y., & Wang, L. (2017). An RFID-enabled positioning system in an automated guided vehicle for smart factories. *Journal of Manufacturing Systems, 44,* 179-190.

Luo, Y., Duan, Y., Li, W., Pace, P., & Fortino, G. (2018). Workshop Networks Integration Using Mobile Intelligence in Smart Factories. *IEEE Communications Magazine, 56*(2), 68-75.

Niu, Y., Lu, W., Chen, K., Huang, G. G., & Anumba, C. (2016). Smart construction objects. *Journal of Computing in Civil Engineering, 30*(4), 04015070.

Peruzzini, M., & Pellicciari, M. (2017). A framework to design a human-centered adaptive manufacturing system for aging workers. *Advanced Engineering Informatics, 33,* 330-349.

Ramaji, I. J., & Memari, A. M. (2016). Product architecture model for multistory modular buildings. *Journal of Construction Engineering and Management, 142*(10), 04016047.

Ren, G., Hua, Q., Deng, P., Yang, C., & Zhang, J. (2017). A Multi-Perspective Method for Analysis of Cooperative Behaviors Among Industrial Devices of Smart Factory. *IEEE Access, 5,* 10882-10891.

Seiger, R., Keller, C., Niebling, F., & Schlegel, T. (2015). Modeling complex and flexible processes for smart cyber-physical environments. *Journal of Computational Science, 10,* 137-148.

Shrestha, P., & Behzadan, A. H. (2018). Chaos Theory–Inspired Evolutionary Method to Refine Imperfect Sensor Data for Data-Driven Construction Simulation. *Journal of Construction Engineering and Management, 144*(3), 04018001.

Taghaddos, H., Hermann, U., AbouRizk, S., & Mohamed, Y. (2012). Simulation-based multiagent approach for scheduling modular construction. *Journal of Computing in Civil Engineering, 28*(2), 263-274.

Yi, W., Chan, A. P., Wang, X., & Wang, J. (2016). Development of an early-warning system for site work in hot and humid environments: A case study. *Automation in Construction, 62,* 101-113.

Wan, J., Chen, B., Imran, M., Tao, F., Li, D., Liu, C., & Ahmad, S. (2018). Toward Dynamic Resources Management for IoT-Based Manufacturing. *IEEE Communications Magazine, 56*(2), 52-59.

Wang, J., Shou, W., Wang, X., & Wu, P. (2016a). Developing and evaluating a framework of total constraint management for improving workflow in liquefied natural gas construction. *Construction Management and Economics, 34*(12), 859-874.

Wang, J., Sun, Y., Zhang, W., Thomas, I., Duan, S., & Shi, Y. (2016b). Large-Scale Online Multitask Learning and Decision Making for Flexible Manufacturing. *IEEE Transactions on Industrial Informatics, 12*(6), 2139-2147.

Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016c). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data-based feedback and coordination. *Computer Networks, 101,* 158-168.

Wang, S., Zhang, C., Liu, C., Li, D., & Tang, H. (2017). Cloud-assisted interaction and negotiation of industrial robots for the smart factory. *Computers & Electrical Engineering, 63,* 66-78.

Wieland, M., Kaczmarczyk, P., & Nicklas, D. (2008, March). Context integration for smart workflows. In Pervasive Computing and Communications, 2008. PerCom 2008. *Sixth Annual IEEE International Conference on* (pp. 239-242). IEEE.

Wu P, Wang J, Wang X. A critical review of the use of 3-D printing in the construction industry. *Automation in*

Construction. 2016;68:21-31.

Wu P, Song Y, Shou W, Chi H, Chong HY, Sutrisna M. A comprehensive analysis of the credits obtained by LEED 2009 certified green buildings. *Renewable and Sustainable Energy Reviews. 2017*;68:370-9.

Zhang, C., & Hammad, A. (2011). Multiagent approach for real-time collision avoidance and path replanning for cranes. *Journal of Computing in Civil Engineering, 26*(6), 782-794.

Zhang, Y., Wang, W., Du, W., Qian, C., & Yang, H. (2018). Coloured Petri net-based active sensing system of real-time and multi-source manufacturing information for the smart factory. *The International Journal of Advanced Manufacturing Technology, 94*(9-12), 3427-3439.

Zhong, Ray Y., Yi Peng, Fan Xue, Ji Fang, Weiwu Zou, Hao Luo, S. Thomas Ng, Weisheng Lu, Geoffrey QP Shen, and George Q. Huang. "Prefabricated construction enabled by the Internet-of-Things." *Automation in Construction* 76 (2017): 59-70.