

Research Article

Colored Petri Net-Based Verification and Improvement of Time-Sensitive Single-Unit Manufacturing for the Soil Preparation Instrument of Space Missions

Kai Leung Yung,¹ Ming Gao ,^{2,3} An Liu,³ Wai Hung Ip,^{1,4} and Shancheng Jiang⁵

¹Department of Industrial and System Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China

²Center for Post-doctoral Studies of Computer Science, Northeastern University, Shenyang, China

³School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian, China

⁴University of Saskatchewan, College of Engineering, Saskatoon, Canada

⁵School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China

Correspondence should be addressed to Ming Gao; gm@dufe.edu.cn

Received 5 June 2020; Revised 17 July 2020; Accepted 20 July 2020; Published 25 August 2020

Guest Editor: Chi-Hua Chen

Copyright © 2020 Kai Leung Yung et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Various space missions, including the Russian and Chinese interplanetary exploration collaboration in 2011 and the Phobos-Grunt space project to be relaunched by the Chinese in 2025, carry a soil preparation system (SOPSY), which is an instrument used for scientific experiments. The design and manufacture of this precision instrument require stringent manufacturing processes and workflow of the highest quality, with every process in the project carefully monitored and controlled. All processes should be completed within the deadline so that the space project can be launched at the scheduled time. The colored Petri net (CPN) modeling method can describe a variety of resource types and execution logic, and it can be formally verified. Based on these advantages, we clearly describe the complex structure of the SOPSY unit production process. In addition, we use critical time and the 6 sigma system to evaluate the availability and reliability of workflows, and we use elimination and simplification (ECSR) methods and constraint theory to improve the manufacturing process of the SOPSY unit. We further provide optimization theories, methods, and insights for workflow management in time-sensitive and independent manufacturing systems.

1. Introduction

Workflow is a case-based term focusing on the logistics part of a business. The purpose of studying it is to ensure that tasks are completed by the right people at the right time [1]. Such workflow has long been a trending topic since the introduction of information systems in the 1990s. At that time, information systems served as the tool to monitor, control, and support the business process [2]. Today, workflow-related research has been performed in a large number of industries; for example, airport traffic simulations [3], service industry [4], distributed manufacturing processes [5], supply chain networks [6], and plastic plants [7]. Although the topic of workflow is discussed in almost every

industry, it has seldom been mentioned in space instrument manufacturing processes, which are extremely complicated and time-sensitive, and must be carefully managed. Hence, a study on workflow simulation analysis for space instrument manufacturing processes is imperative.

Petri net (PN) is a simulation tool developed by Dr. Carl Adam Petri in his doctoral paper. The classic Petri net, which often refers to the original one, has three elements: place, transition, and functions [8]. Mathematically, a Petri net can be represented with a tuple (P, T, F) . P stands for a finite set of places, T stands for a finite set of transitions, and F stands for a finite set of functions. Places and transitions map to buffers and activities. Functions involve the logical relationship between places and transitions, and tokens that

flow in the net can represent the state of the system. When dealing with a complex system, the classical Petri net is often insufficient for modeling such a system. The colored Petri net (CPN) introduced a new element—color—to the classical PN [9]. The attributes of token can now be specified through color, which helps exponentially enlarge the complexity of the system that CPN can model. As a fully developed modeling tool, CPN has been used in many industries, but its application in the space instrument manufacturing process is limited.

Phobos-Grunt is a Russian space mission that launched in 2011. However, the mission was not successful. One of its systems is the soil preparation system, which was developed by Hong Kong Polytechnic University [10]. The space mission is planned to re-fly in 2025, and it is, therefore, necessary to examine every process of the mission and ensure that the failure will not re-occur. This case, therefore, provides an opportunity to conduct research on the spacecraft manufacturing process workflow management using CPN.

This paper focuses on analyzing the workflow of a space instrument manufacturing process prior to the implementation stage through CPN modeling. It is imperative to build an appropriate workflow model at the abstract level in order to support process design and analysis [11]. Thus, the first focus of this paper is to present an appropriate CPN model for a specific space instrument manufacturing process. This model should be appropriate, comprehensive (considering both schedule and resources), and readable. Secondly, for the purpose of presenting a workflow improvement method, this paper will use the data derived from the case colored Petri net simulation to propose a new workflow using an existing project scheduling model. Then, another CPN model will be constructed to validate the performance of the proposal. The aim is to complete in time and improve the resource utilization.

This paper will contribute with respect to both practical and academic contexts. The practical contribution of this paper would be the optimization of the Phobos-Grunt SOPSYS delivery unit manufacturing process. The Phobos-Grunt space project is scheduled to relaunch in 2025. There are less than five years left to prepare for the relaunch; hence, every process in the project should be carefully monitored and controlled. All processes must be completed within the deadline so that the space project can be relaunched at the scheduled time. The academic contribution of this paper is that it has reference value for other time-sensitive unit manufacturing process optimization methods. The CPN modeling method we use can represent more complex manufacturing processes. For example, the CPN model used in this paper can clearly represent the delay time and rework structure. In addition, CPN's existing simulation software is more integrated, and the model can be formally verified [12]. This is something that traditional DAG and AOA cannot do.

2. Related Work

2.1. Workflow Optimization. Regarding the research of the workflow optimization method, Dewan et al. [13] focus on

the task redesign process in workflow optimization. The study uses the case of IBM business information flow to demonstrate the technique. Dewan et al. argue that job bundling could be a useful approach to improve communication between tasks. This point is quite useful in this paper because during manufacturing processes, the rework delay can be effectively reduced through good communication. This is an example for task redesign. Another study focuses on the workflow optimization approach itself and presents a way to use FlowOpt in workflow optimization processes [14]. This process includes modeling, visualizing, analyzing, and optimizing the production workflow. Similarly, our paper will follow in the case modeling and analysis using software optimization of the workflow. Crop et al. also conducted a study on workflow optimization in the radiotherapy industry [15].

Studies in workflow optimization have been conducted in different industries. Dewan [13] optimized the workflow of business information flow. Karlik et al. [16] used CPN modeling and simulation methods to study the time and throughput of data-aware workflows to reflect the actual workload and gross profit of an enterprise over a period of time. Yanhua Du et al. [17] used Petri nets and neural networks in the development of an intelligent logic controller for an experimental manufacturing plant to provide the flexibility and intelligence required for this type of dynamic system. Kasemset [18] optimized the manufacturing process in a paper package factory. However, one industry has rarely been mentioned in the study of workflow optimization, the aerospace industry. Essentially, there is no comprehensive workflow optimization study that has been conducted in the aerospace industry.

Various qualitative and quantitative methods have been used to study workflow improvement. One paper package manufacturing process study used the ARENA program to model the workflow and evaluated the performance [18]. This is a sound approach in verifying the credibility of the optimization proposal, but the model built is too simple to reflect a real-world case. The study conducted by Dewan et al. [13] uses a quantitative validation method. It developed its own algorithm to calculate the delay per unit and cost per delay time unit. The algorithm is correct to some degree, but it still works in a relatively abstract manner that is rather difficult to understand. Crop et al. used Python-based discrete event simulation software to simulate the process and validated its ConWIP-based proposal [15]. The basic concept of this software is to logically model the process from a resource and time perspective. It is a reliable simulation tool but is not suitable for modeling concurrent systems.

In a study conducted by Lv et al. [5], the authors use a hierarchical CPN to model a distributed manufacturing process. In this study, Petri net's properties make it possible to simulate the concurrency and synchronization of a system. Hence, the hierarchical modeling approach can be achieved through a hierarchical CPN. This modeling technique is implemented in this paper.

Another common example of Petri net is in supply chains. Wang et al. presented a Petri net implementation method in a proactive resilient holistic supply chain network

[6]. It presents the specific CPN structure built for the supply chain network. This is a solid approach to show the establishing process of a CPN model. Although the study provides a highly detailed design process of the modeling, it is not validated through specific case data.

A CPN implication in a plastic manufacturing process workflow management was conducted by Fung et al. [7]. Their study placed substantial effort into describing how to map the workflow elements into the CPN model. This is also the main focus of this study. Although this study presents a highly detailed mapping theory regarding how to use CPN elements to represent workflow elements, there are no data to demonstrate that the model can actually generate real data, such as delay time or resource utilization.

A CPN simulation has been implemented in an airport pavement traffic case [3]. That study used the subnet concept to establish the CPN model for the case. It first introduced the concept of CPN and other extended CPNs, and then the model for the case was developed. The simulation data were then collected; although the simulation data showed a certain degree of flight delay, the study did not continue to optimize the workflow.

We have reviewed the workflow optimization literature related to this paper. A gap in the research area has been identified: there is no workflow optimization study for aerospace industry cases. Second, mixed qualitative and quantitative workflow optimization methods have seldom been demonstrated through simulations or algorithms. Third, we have summarized some effective viewpoints about the realization of CPNs in workflow optimization research, and we determined that discrete event simulation software that can simulate concurrent systems should be the best choice for modeling and analysis.

In this paper, we first introduced the basic concept of the CPN and extended CPNs. Then, we explained the mapping theory to map case workflow elements into CPN elements: the CPN model will be developed with a bottom-up approach that involves constructing the entire process from the substructure. Finally, we will collect and analyze several key data in the manufacturing process and propose improvement measures. The contribution of this study is that we make use of the relatively integrated software support of CPN, the color set of the theoretical model, time modeling and automatic verification, and other advantages to propose and verify improvement suggestions through case studies. Furthermore, the logic of this process is similar to the manufacturing process of most time-sensitive single units, and the same process and technology can be used in other manufacturing systems to achieve workflow optimization.

3. Methodology

The focus of this article is to use the CPN Tools simulation optimization method to optimize the manufacturing process of the SOPSYS delivery unit and provide a reference method and framework for such process optimization relating to space missions. The method and framework are shown in Figure 1. First, we sorted out the parameters involved in the problem and used CPN Tools to build an initial model

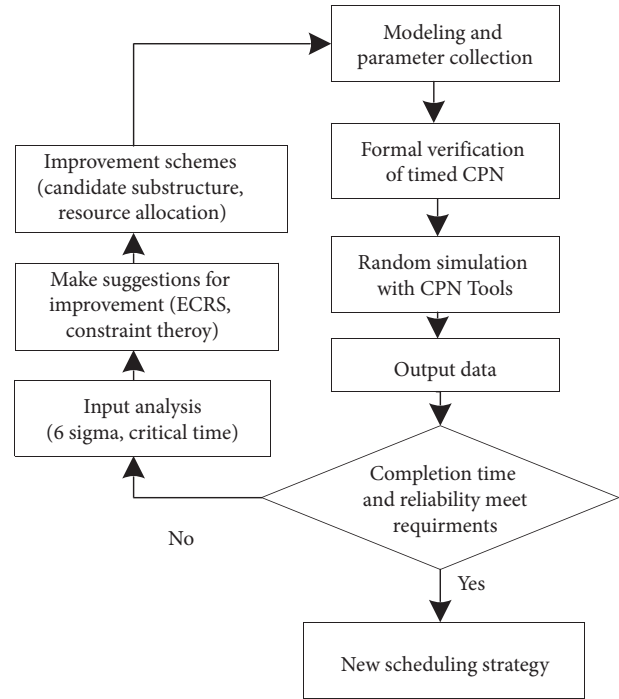


FIGURE 1: Research process.

according to the original scheme. We then simulate the original model through the CPN Tools and extract the key data in the simulation results; we use the key time analysis method and the 6 sigma criterion to analyze the simulation data; finally, we use the ECSR method and constraint theory to obtain improvements. This process is repeated until the final scheduling strategy and resource allocations that meet the project deadline.

3.1. CPN Theory and Formal Verification. Previously, we illustrated that PN has been successfully used for modeling [19]. Compared with DAG, CPN can express more complicated process, such as or-split selection paths, and-join parallel path, and defined relationships of resource dependency and consumption and the characteristics of space-time networks. Besides, the software support of CPN is comparatively integrated; it supports substructure modeling, color sets of theoretical models, time modeling, and automatic validation.

3.1.1. Petri Net Theory. To facilitate understanding, we have summarized the meaning of the symbols in Table 1.

Various types of extended PN are defined as follows:

- (1) A PN is a 5-tuple, $PN = (P, T, F, W, M_0)$ where the PN structure $N = (P, T, F, W)$ without any specific initial marking is denoted by N .
- (2) A CPN, $R = \{P, T, C, I^+, I^-, M\}$ where an item of $C(s)$ is called a color of s and $C(s)$ is called the color set of s .
- (3) TPN was developed based on traditional PN. In TPN (timed Petri net), each transition takes a “real time”

TABLE 1: PN element interpretation.

PN element	Explanation
P	$P = \{p_1, p_2, \dots, p_m\}$, a finite set of places
T	t_1, t_2, \dots, t_n , a finite set of transitions
F	$F \subseteq (P \times T) \cup (T \times P)$, a set of arcs (flow relations)
W	$\rightarrow \{1, 2, 3, \dots\}$, a nonempty and finite set of weightings for arcs
M_0	$P \rightarrow \{1, 2, 3, \dots\}$, $P \cup T \neq \emptyset$, $P \cap T \neq \emptyset$, initial marking
C	A colored function from $P \cup T$ to W
I^+, I^-	Forward (backward) incidence matrix of $C \times T$, where $I^+(p, t)$ is a function from $C(p) \times C(t)$ to rmN , where N is the set of nonnegative integers
M	Initial marking of the net which is a vector of P , where $M(p)$ is a function from $C(p)$ to N
f	Firing time function, which assigns the nonnegative (average) firing time $f(t)$ to each transition t of the net
n	The number of classes that model the system dynamics
R	Priority index for different products or jobs
V_i	The set of variables in NHTCPN

to fire (i.e., there is a firing time associated with each transition of a net that determines the duration of the firing of a transition [20]).

A conflict-free TPN is a pair $N_{\text{timed}} = (M, f)$, where M is a conflict-free marked PN, $M = (N, M_0)$, $N_{\text{timed}} \rightarrow R^+$, and R^+ denotes the set of nonnegative real numbers [21].

- (4) A timed CPN (TCPN) is a structure defined as $N = (P, T, F, W, M_0, f, R)$, which is developed based on the traditional PN, CPN, and TPN; hence, it should contain all of the elements of these three existing PNs. Besides that, a new element, priority for different product or job R , is introduced for the new PN in (1). In real cases, different products or different production lines will adjust the processing sequence based on its current significance, such as customer-wanted date or the importance index of the customers.
- (5) Hierarchical TCPN (HTCPN) models the dynamic evolution of a system [22]. It is composed of a set of HTCPN subnets. The definition for the HTCPN is given as follows:
- (a) An HTCPN is composed of a finite set of HTCPN subnets (i.e., $\text{NHTCPN} = \{\text{NHTCPN}_1, \text{HTCPN}_2, \dots, \text{NHTCPN}_l\}$, where l is the number of classes that model the system dynamics.)
- (b) $\text{NHTCPN}_i = \{-A_i \text{NHTCPN}_i, V_i, M_{0i}\}$, where A_i is the name of the subnet and NHTCPN_i is a TCPN (following (4))

The definition of the HTCPN subnet is based on the identity structure of a certain network. In a complex hierarchical system, when modeling from a holistic perspective, there may be multiple alternate transitions, and each is associated with a complex HTCPN termed a subnet. Subnets allow a more specific and accurate interpretation of the activities in the replacement transformation. Each subnet can be thought of as a unique integrated HTCPN, but its inputs and outputs are closely tied to other subnets. Therefore, HTCPN can reduce the complexity of modeling and improve readability.

3.1.2. Petri Net Properties. For the verification of the manufacturing process model, some properties and analysis methods of the basic Petri net are used, among which the properties involve reachability, boundedness, activity, and fairness. A common analysis method is to construct the reachability identification map of the Petri net, which can be used to analyze the state of the network system and the sequence of its occurrence in order to know the relevant properties of the network system. Finally, rules based on these properties are specified to achieve the purpose of automatic formal verification. The natural and formal verification theorems are as follows:

Definition 1. Reachability of the Backbone Process. Under the network identifier M_0 , the only execution token E is in the starting place of the process. After the limited sequence of transitions, the network identifier arrives, and the only execution token E is located in the last place of the process.

Definition 2. Reachability of Task Subprocess. In contrast to the main process, one or more execution tokens E are in the starting place of the process in the network identifier M_1 . After a limited sequence of transitions, the network identifier M'' arrives, and all the execution tokens E are in the ending place of the process. As a result, the subprocess of tasks supports the multithreaded execution, and this subprocess can simulate parallel and serial execution of multiuser, multirole, and multitask.

Definition 3. Activity. After adding a virtual line (transition and the color of the input arc and output arc is the execution token) between the place where the process starts and ends, for any two libraries P_1 and P_2 , if the execution token E is in P_1 , after a limited transition sequence, E reaches P_2 . That is, only the CPN of the color set E is examined, at which point it is degraded to the basic Petri network, which is verified by the definition of its activity.

Definition 4. Fairness. After adding a virtual line (transition and the color of the input arc and output arc is the execution token) between the beginning place and the ending place of

the process, after a limited transition sequence, each place has the opportunity to receive an execution token.

Definition 5. Reachable Marked Graph. The reachable identification set $R(M_0)$ of a bounded Petri net is a finite set. $R(M_0)$ can be used as a vertex set, and a direct reachability relationship between the identifiers is used to form a directed graph. This graph is termed a Petri net's reachable marking graph, and it can be used to analyze the state changes and transition occurrence sequences of the network system in order to know the relevant properties of the network system.

Definition 6. Formal Verification Rules for Workflow.

Rule (1): stimulation of changes and changes in multiple sets of color examples to the place follow HTCPN.

Rule (2): in the initial state M_0 , the only execution token E is in the starting place of the trunk process; when the trunk process stimulates parallel transitions, the unique execution token copies an execution token for each branch, which is referred to as explicit defined parallelism; when other types of control transitions (serial, select, or loop) are fired, the unique execution token remains unchanged and only behaves as a pass.

Rule (3): for task subprocesses, the operating mechanism of serial, selection, or cyclic control changes is the same as Rule (2). Parallel control changes need to be classified and discussed.

Explicitly defined parallelism between a single execution token, as in Rule (2); the implicit parallelism between multiple execution tokens is a single branch of the CPN graphic definition repeatedly. With the multiple execution tokens, the same transitions are triggered multiple times simultaneously. Dynamic, on-demand simultaneous creation of multiple execution tokens is involved, creating tasks in parallel; the serialization of multiple execution tokens is a single branch of the CPN graphic definition. With multiple execution tokens, the same transitions are triggered in a

controlled manner multiple times. This approach creates multiple execution tokens dynamically, on demand, and serially.

3.1.3. Steps for Modeling Method with HTCPN

Step 1. Outline the overview structure of the object-distributed manufacturing network, and transform all the elements into the above CPN elements to build up first-level PN.

Step 2. Further define the subnet for the first-level HTCPN.

Step 3. Check whether the subnet meets the above definition and verification rules, and the subnets are always the substitution for transition element of the last level. If it meets the requirements, go to Step 5; otherwise, go to Step 4.

Step 4. Modify the PN model structure to make it conform to all definitions and rules, and go to Step 3.

Step 5. Repeat Step 2 until the transition elements are not required for further development, which means that the basic elements in a distributed manufacturing network have been modeled.

Step 6. Ensure that all of the links between the levels are consistent through the entire HTCPN.

3.2. Optimization Mathematical Model. According to the steps of HCPN modeling in the previous section, if the workflow is effective, it must meet the definition and rules of PN. Next, we need to calculate the workflow completion time of the current iteration under the condition that the workflow is valid (meets the latest deadline and is reliable), and we express the calculation process by means of a mathematical model.

The variable interpretation is shown in Table 2.

The mathematical model is as follows:

$$\text{Min}(T^{\text{SD}}), \quad (1)$$

$$T^{\text{Avg}} \leq D, \quad (2)$$

$$T_{ijk}^s \geq T_{ijk}^{\text{ESD}}; \quad i = (0, 1, 2, \dots, n), j = (1, 2, 3, \dots, \mu_i), k = (1, 2, 3, \dots, \sigma_{ij}), \quad (3)$$

$$T_{ijk}^f = T_{ijk}^s + T_{ijk}^e; \quad i = (0, 1, 2, \dots, n), j = (1, 2, 3, \dots, \mu_i), k = (1, 2, 3, \dots, \sigma_{ij}), \quad (4)$$

$$T_{ijk}^f \leq T_{ijk}^{\text{LED}}; \quad i = (0, 1, 2, \dots, n), j = (1, 2, 3, \dots, \mu_i), k = (1, 2, 3, \dots, \sigma_{ij}), \quad (5)$$

$$T_{\text{critical}}^i = \text{Max}_{\substack{j \in (1, 2, \dots, \mu_i) \\ k \in (1, 2, \dots, \sigma_{ij})}} (T_{ijk}^f), \quad i = (1, 2, \dots, n), j = (1, 2, 3, \dots, \mu_i), k = (1, 2, 3, \dots, \sigma_{ij}), \quad (6)$$

TABLE 2: Mathematical variable interpretation.

Symbol	Description
n	The number of workflows
W	The set of workflows $W = \{W_1, W_2, \dots, W_n\}$
W_i	The i th workflow in the workflow set
μ_i	The number of substructures in the workflow W_i
V_i	The set of substructures in workflow W_i , $V_i = \{V_{i1}, V_{i2}, \dots, V_{i\mu_i}\}$
V_{ij}	The j th substructure of workflow W_i
σ_{ij}	The number of tasks of substructure V_{ij}
U_{ij}	The task set of substructure V_{ij} , $U_{ij} = \{U_{ij1}, U_{ij2}, \dots, U_{ij\sigma_{ij}}\}$
U_{ijk}	The k th task in the substructure U_{ij}
P_{ijk}	Predecessor set of task U_{ijk}
D	Deadline for the project
m^f	The number of resource types
$\vec{R}^f(t)$	The resource vector owned at the current time t $\vec{R}^f(t) = \{R_1^f(t), R_2^f(t), \dots, R_m^f(t)\}$
$\vec{R}(i, j, k)$	Resource vector required for task U_{ijk} execution $\vec{R}(i, j, k) = \{R_1^r(i, j, k), R_2^r(i, j, k), \dots, R_m^r(i, j, k)\}$
T_{ijk}^e	Time required for task U_{ijk} to execute
T_{ijk}^s	Time when task U_{ijk} started to execute
T_{ijk}^f	Time when task U_{ijk} execution ends
T_{ijk}^{ESD}	The earliest start time of the task U_{ijk} (calculated according to the project start time)
T_{ijk}^{LED}	The latest end time of the task U_{ijk} (calculated according to the project deadline)
$T_{\text{critical}w}^i$	Time to complete the critical path of workflow W_i
T^{Avg}	Average completion time of all workflows
T^{SD}	Standard deviation of completion time

$$x_{ijk}(t) = \begin{cases} 1, & \text{Task } U_{ijk} \text{ executed at time } t, \\ 0 & \text{Otherwise,} \end{cases} \quad i = (0, 1, 2, \dots, n), j = (1, 2, 3, \dots, \mu_i), k = (1, 2, 3, \dots, \sigma_{ij}), \quad (7)$$

$$\sum_{i=1}^n \sum_{j=1}^{\mu_i} \sum_{k=1}^{\sigma_{ij}} (t) \cdot \vec{R}^r(i, j, k) \leq \vec{R}^f(t), \quad (8)$$

$$T_{ijk}^s \geq \text{Max}_{k' \in P_{ijk}} (T_{ijk'}^f), \quad i = (0, 1, 2, \dots, n), j = (1, 2, 3, \dots, \mu_i), k = (1, 2, 3, \dots, \sigma_{ij}), \quad (9)$$

$$T^{\text{Avg}} = \frac{\sum_{i=1}^n T_{\text{critical}}^i}{n}, \quad (10)$$

$$T^{\text{SD}} = \sqrt{\frac{\sum_{i=1}^n (T_{\text{critical}}^i - T^{\text{Avg}})^2}{n}}. \quad (11)$$

Equation (1) represents the objective function of the mathematical model. The purpose of using the minimized standard deviation as the target value is to provide a basis for the reliability analysis in the following section; equation (2) indicates that the average value of all workflow simulation results must be less than the project deadline; equation (4) is used to calculate the completion time of the task, and equations (3) and (5) define the time window constraint of the task U_{ijk} ; that is, the start time of the task is later than the earliest start time of the task and the end time. It must be earlier than the latest end time of the task. Among them, T_{ijk}^{ESD} and T_{ijk}^{LED} can be calculated according to the start time and deadline of the space project, and we will explain this in detail in the experimental analysis section. In equation (6), we use the

concept of completion time of the critical substructure consistent with the following sections to represent the completion time of the entire workflow; in equation (7), $x_{ijk}(t)$ represents the decision variable, and when the task U_{ijk} is executed at time t , $x_{ijk}(t)$ is equal to 1; otherwise, $x_{ijk}(t)$ is equal to 0; the resource limit is defined in equation (8) and represents the resources required for task execution at any time that cannot exceed the resources currently owned; equation (9) expresses the timing constraint relationship between tasks—the current task can only be executed after the execution of all previous tasks is complete; equations (10) and (11) represent the calculation process of the average and standard deviation of workflow completion time, respectively.

3.3. *Analysis Methods.* In this section, we will introduce some methods that will be used in the simulation analysis process.

3.3.1. *6 Sigma System.* To ensure the reliability of the manufacturing process, the 6 sigma system was introduced to the simulation iteration.

The 6 sigma model is shown in Figure 2. The USL and LSL are the limits of this system. If the result is normally distributed, the distribution diagram is within the acceptable range of plus or minus three sigma, which is the standard deviation [23], and the results located outside the area will be 0.135% [24]. Hence, if the company wishes to maintain the service level target at 99.865%, the data derived from the case CPN simulation require a new workflow; the 6 sigma system was introduced to the simulation iteration to control the completion time before the project deadline. 6 sigma has been demonstrated to be an effective management framework and methodology for improving operation workflows in industry [25, 26]; various quality systems in engineering and service industry use this systematic approach to implement quality standards for process control and monitoring. Our case study of space instruments, hence, makes use of this strategy to improve the detailed operations, from simulation to implementation of the quality management system [27, 28]: it follows DMAIC (Define, Measure, Analyze, Improve, and Control). Define, Measure, and Analysis are carried out through simulations of workflow changes described in Section 6.1, and Improve and Control are performed with elimination and simplification following Control of critical resources and allocation of assemble, rework, and inspection tasks, as illustrated in Section 6.2.

3.3.2. *Critical Time Analysis.* During the initial design phase of the manufacturing process, each task is assigned to the theoretically earliest start date (TESD), which is the earliest time that a task can begin without considering resource constraints. The start time of each task cannot be earlier than TESD because some tasks need to wait for the predecessor to complete execution. The TESD of each task can be calculated according to equation (12), and the LT_n represents the execution time of task n :

$$TESD_{n+1} = TESD_n + LT_n. \quad (12)$$

The latest start date (LSD) is another important data point to analyze. The LSD of the task is calculated forward according to the deadline for the entire process. If the task cannot be initiated before the latest start time, the entire system may not be completed within the specified deadline. The LSD of each task can be calculated according to the following equation:

$$LSD_n = LSD_{n+1} - LT_n. \quad (13)$$

Through the TESD and LSD of each task, we can calculate the time window of each task, which is $LSD - TESD$. If there is a task in the simulation results that cannot be completed within the time window, the task is the key object

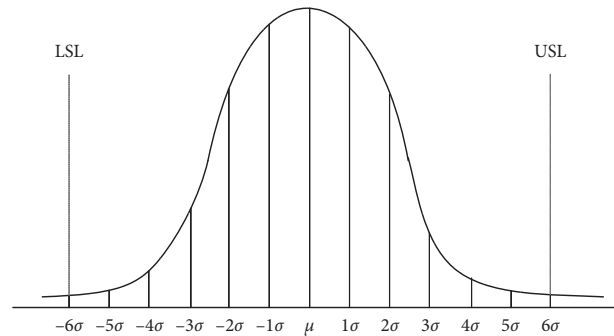


FIGURE 2: Six sigma.

to be analyzed. The specific situation will be introduced in detail in Section 5.

3.3.3. *ECRS Method.* The ECRS method is one of the most fundamental activity processing tools in industrial engineering [18]. The letter *E* stands for elimination of unnecessary work, *C* for combination of operations, *R* for activity sequence rearrangement, and *S* for necessary work simplifications. These four principles have been implemented in many industries. The ECRS method has been used in ice cream manufacturing processes [29], electronic manufacturing industries [30], etc.

The ECRS method will be the base principle for this improvement proposal. For every critical issue, this dissertation will attempt was proposed by Dr. Goldratt in 1990 [29]. This theory was developed for optimized production technology, focusing on identifying the constraints inhibiting the successful processing of the workflow and then solving them. The theory of constraints lists eight rules for manufacturing process planning:

- (a) Balance logistics or material instead of balance capacity.
- (b) Noncritical resource utilization is decided by the system constraints instead of its own potential capacity.
- (c) High resource utilization does not necessarily mean that the system is working in an efficient manner.
- (d) Delays in critical path usually mean delays to the entire system.
- (e) Saving time or resources in noncritical paths often has no positive impact on the system.
- (f) Critical paths normally will decide most properties of the system, for example, the system lead time, system productivity, and inventory.
- (g) Batch size should vary with time.
- (h) Priority should be designed based on the system constraints. Early finished tasks are not resulted from the initial design but the processing of the plan.

Those eight rules mention a particularly important improvement criterion and the critical path, and provide substantial inspiration for the improvement of our workflow

structure. We will elaborate upon specific improvement measures in the experimental result analysis section.

4. Case Modeling and Simulation

4.1. Case Description. In the case of application, the space mission is to explore Phobos, which is one of the two moons that orbit around Mars. The Phobos-Grunt mission planned to land on Phobos to retrieve samples for the purpose of scientific research and also to serve as an illustration for future sample-retrieving space missions [31]. The mission failed because the spacecraft never managed to leave the earth's orbit [32]. The second attempt of the Phobos-Grunt mission is planned in 2025 with minor amendments, which makes the research focus of this paper not only illustrative, but also practical. The architecture of the Phobos-Grunt soil preparation system (SOPSYS) to be landed on Phobos is shown in Figure 3 [10]. The actuator E will first deliver the sample particles inside the spacecraft for analysis and then deliver the debris outside the spacecraft into space. The study focus is Part E 's manufacturing process.

The delivery unit E consists of five parts, and the code name for the parts in its manufacturing process is PART0030 (delivery motor with encoder), PART0031 (sieve encoder PCB), PART0032 (sieve encoder receiver), PART0033 (sieve encoder LED), and PART0034 (delivery chamber). One subpart of CPN0012 (delivery encoder) is assembled with PART0031 and PART0032. The delivery units deliver the sample particles into the spacecraft and deliver debris into space.

There are five parts that must be purchased or manufactured throughout the process; the sequence is shown in Figure 4. There are 5 engineers, 4 operators, and 1 inspector in this process. The entire process begins with the initial design of CPN0005 (the Phobos-Grunt SOPSYS delivery unit). After the CPN0005 is designed, 4 tasks can then start to proceed simultaneously: the detailed specification of PART0030 and PART0033, the design and budgeting of CPN0012, and the design, budgeting, and test equipment design of PART0034. Purchasing PART0030 and PART0033 comes after the detail specification, and purchasing PART0031 and PART0032 comes after the design of CPN0012. PART0034 is the only part that needs to be in-house manufactured. The manufacturing of PART0034 will begin immediately after its design phase. After every purchasing, manufacturing, and assembly process, there will be an inspection process to ensure its quality. Hence, when PART0030, PART0031, PART0032, and PART0033 are received and PART0034 is manufactured, they need to be inspected to proceed to the later assembly process. After PART0031 and PART0032 are inspected, CPN0012 will then be assembled, and another inspection comes after it. Finally, PART0030, CPN0012, PART0033, and PART0034 will be assembled together into CPN0005, which will be tested and inspected. The network flow of the entire original manufacturing process and required resources are shown in Table 3.

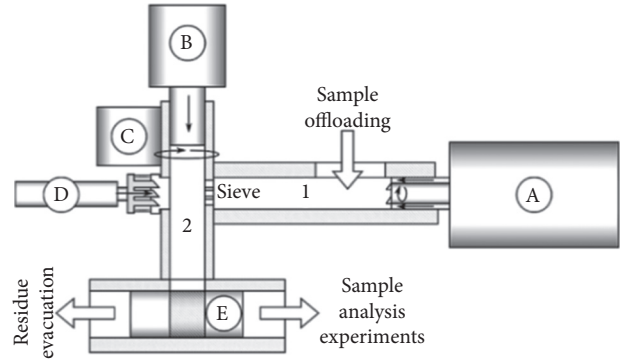


FIGURE 3: Architecture of SOPSYS.

4.2. Mapping Workflow Elements into CPN. In this section, we will describe the basic workflow structure of the HCPN [33], thus formalizing the SOPSYS process model. First, we mapped between the Phobos-Grunt SOPSYS delivery unit manufacturing process case and CPN, as shown in Table 4.

According to the description of (1) in the first part of the Methodology section, $P = \{p_1, p_2, \dots, p_m\}$, with a finite number of places (nodes). Places in this CPN can be imagined as a warehouse between different tasks that holds the material along the manufacturing process and the staff pool that provides human resources. Two different places do not necessarily have to be two different physical locations because places are defined by the tokens they are holding.

$T = \{t_1, t_2, \dots, t_n\}$ is a finite transition that represents the dynamic process of advancing from one node to the next, due to certain conditions or rules being met. By defining the tasks as a transition, we can naturally model the resources and products as tokens in the CPN. Tokens in the CPN can represent the resources and products consumed and produced along the PN. For example, for the task PART0032, the inspection requires one inspector and one uninspected PART0032 to produce an inspected PART0032. Two tokens are the prerequisite for this transition, one represents the inspector, and one represents the uninspected PART0032. In the firing process of this transition, these two tokens are consumed, and two new tokens are generated. Consequently, one token representing the inspected PART0032 will be ready for its next transition, and another token representing an idle inspector will be generated for its next transition.

In the SOPSYS model, different colors represent various types of things; therefore, the color can be the thing it stands for. In this case modeling, there are four colors in total. Each color represents a type of resource or token. The three colors are color engineer, color operator, and color inspector.

$f = (\text{REFT} : T \rightarrow \text{IN}, \text{RLFT} : T \rightarrow \text{IN} \cup \infty)$ represents functions describing the relative earliest and the relative latest firing times of the transitions, where clearly $\text{REFT}(t) \leq \text{RLFT}(t)$ for each $t \in T$, where IN is the set of naturals (including 0). In the context of SOPSYS, the relative earliest firing time and the relative latest firing time of the transitions are firing domains of enabled transitions. The definition of REFT and RLFT differs on transitions with different colored types [14].

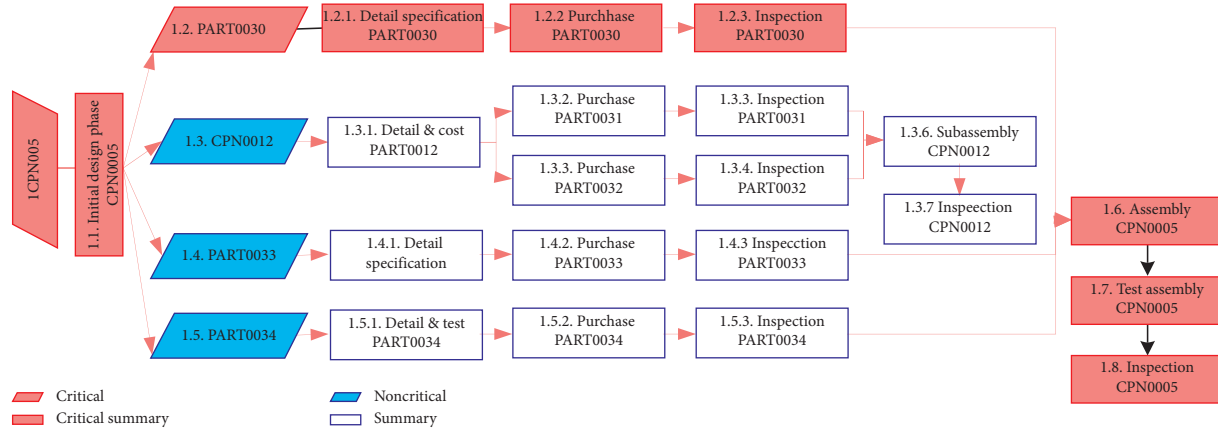


FIGURE 4: Manufacturing process.

TABLE 3: Manufacturing process and required resources.

ID	Task name	Duration (days)	Predecessor	Resource name (number)
1	1. CPN0005	100		
2	1.1. Initial design phase CPN0005	40		Designer (2)
3	1.2. PART0030	38		
4	1.2.1. Detail specification PART0030	5	2	Designer (2)
5	1.2.2. Purchase PART0030	30	4	
6	1.2.3. Inspection PART0030	3	5	Inspector (1)
7	1.3. CPN0012	34		
8	1.3.1. Design & cost CPN0012	7	2	Designer (2)
9	1.3.2. Purchase PART0031	21	8	
10	1.3.3. Purchase PART003	5	8	
11	1.3.4. Inspection PART0031	1	9	Inspector (1)
12	1.3.5. Inspection PART0032	3	10	Inspector (1)
13	1.3.6. subassembly CPN0012	2	11, 12	Operator (1)
14	1.3.7. Inspection CPN0012	3	13	Inspector (1)
15	1.4. PART0033	23		
16	1.4.1. Detail specification PART0033	5	2	Designer (2)
17	1.4.2. Purchase PART0033	15	16	
18	1.4.3. Inspection PART0033	3	7	Inspector (1)
19	1.5. PART0034	21		
20	1.5.1. Design & test CPN0034	8	2	Designer (2)
21	1.5.2. Manufacturing PART0034	10	20	Operator (2)
22	1.5.3. Inspection PART0034	3	21	Inspector
23	1.6. Assembly CPN0005	8	16, 14, 18, 22	Operator (4)
24	1.7. Test CPN0005	7	23	Inspector (4)
26	1.8. Inspection CPN0005	7	24	Inspector

TABLE 4: Mapping between case and CPN.

Workflow element	CPN element
Tasks	Transitions
Resources and products	Tokens
Types of resources and products	Colors
Warehouse and staff pool	Places

HTCPN = $(A_i, NTCPN_i, V_i, M_0)$ is an HCPN model that characterizes the hierarchical model of the SOPSYS. This paper will construct a CPN model using a bottom-up approach. The subprocess will be modeled as a subnet, and the subnet will be drawn as a transformation in the CPN

overview [34]. In this case, the HCPN does not apply to resources being modeled into the CPN; therefore, it is difficult to find a single port subnet. However, developing a model using a bottom-up approach can improve the readability of the CPN. The first step is to build a simple CPN for different subprocesses.

There are five parts in total. Two parts have the same process and can be described in Figure 5.

As shown in Figure 5, PART0030 and PART0033 both begin from a place name P1 (different places in the final CPN). The first transition is specification, which consumes tokens from P1 and Engineer. Token in P1 stands for the status where PART0030 or PART0033 is ready to be

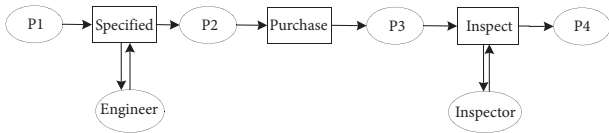


FIGURE 5: Subprocess of PART0030 and PART0033.

specified. Token in Engineer stands for engineers that are available for the task. After the transition specified is fired, new tokens will be generated into P2 and Engineer. Newly generated tokens in P2 have the same color set as the tokens consumed from P1, which stands for the status of the product. Newly generated tokens in Engineer have the same color set as the tokens in Engineer, which stands for available engineers. Similarly, transition inspections consume product status color tokens from P3 inspector color token from place Inspector and generate product status color token into P4 and inspector color token back to place Inspector.

Two other parts also share the same process, which is shown in Figure 6.

PART0031 and PART0032 begin immediately after the design of CPN0012, and therefore, the only difference here is that this process does not have a specification transition.

The only special process is PART0034. The process of PART0034 is shown in Figure 7.

As shown in Figure 7, PART0034 is in-house manufactured, which requires operators in the manufacturing process. The operator color token will be consumed from the place Operator, and the product status color token will be consumed from P2. New operator color tokens will be generated back to the place Operator, and new product status color tokens will be generated into P3 for further processing.

The process of CPN0012 can be modeled as follows.

Figure 8 shows the subnet for the process of CPN0012. There are two aspects to be noticed in the subprocess of CPN0012. The first is that P2 now provides tokens for both PART0031 and PART0032, and these two transitions can be fired concurrently. Thus, P2 must have two tokens in place to trigger both transitions. The next aspect is that two places of Inspector in Figure 7 are the same and contain only a limited number of inspectors. In this case, there is only one inspector available.

There are four places at the beginning of the middle stream to differentiate the four processes; otherwise, one of the transitions might be fired multiple times and other transitions will not be fired at all. If the tokens standing for the products status are divided into four types, which can represent four different subparts, then a more readable CPN can be established, as in Figure 9. The transition subpart is a complicated subnet containing all processes of the four parts. It is plotted as a transition to increase the readability of the CPN, but it is not a single port subnet, and therefore, it cannot be simulated in this manner.

4.3. CPN Tools Modeling. CPN Tools is a software tool for editing, simulating, and analyzing CPNs [35]. CPN Tools can simulate the performance of the model in real time. The declaration in CPN Tools defines the data type in the CPN model [36]. The declaration of the CPN Tools model is shown

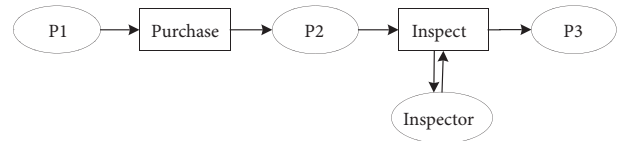


FIGURE 6: Subprocess CPN for PART0031 and PART0032.

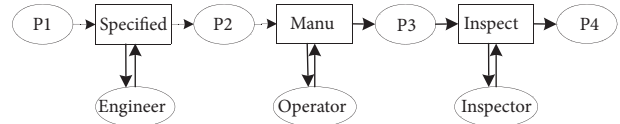


FIGURE 7: Subprocesses CPN of PART0034.

at the bottom right of Figure 10. When modeling a CPN into CPN Tools, the structure of the net must be modified. It is necessary to model the CPN with the computer logic. The structural modification includes the construction of the following parameters: real-time element, setting place color set, rework structure, and construction of output transition.

Real-Time Elements. The real-time element is integrated into the CPN Tools model through adding a time lag to transitions. Figure 10 shows how to achieve this.

The syntax for implementing real-time is $@ + (\text{time})$. Figures 4–10 indicate that transition Inspect will take 7 units of real time to fire. If the timed token consumed in this transition is (elements in a color set) @t, the newly generated token would be (elements in a color set) $@(t + 7)$.

Define Color Groups. A place can only hold certain types of tokens. In the left-hand side of Figure 10, the text “inspector” in the lower-right corner indicates its color setting. If there are multiple types of tokens in one place, their color set should be specifically defined as a list of token types. If a place holds more than one type of token, its color set should be specially defined as lists of token types. A list is a tuple containing variables from different color sets. For example, in this case, if a place represents a pool of staff for all three types of worker (engineers, operators, and inspectors), the elements in the color set here are $(e, O, \text{and } I)$.

As shown in the left-hand side of Figure 10, the task before an inspection will randomly generate an integer token with a value ranging from 1 to 100. The model assumes that all these tasks will have a scrap rate or rework rate of 5%. Thus, the code segment of the inspection transition will use value 5 as the differentiator of a good product and a bad product. If the value is less than or equal to 5, then a token with color bool value false will be generated into the next place. This will not trigger anything further ahead. In the meantime, another token with color value true will be generated into the rework place. This token will trigger the rework transition, which will send the M token back before the task before inspection. If the randomly generated value is greater than 5, then the process moves forward as expected.

An additional transition must be established in the CPN to export the performance data. The transition is shown at the

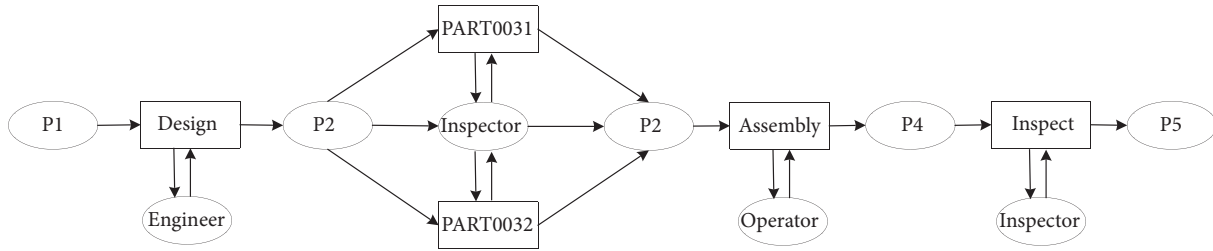


FIGURE 8: Subprocess CPN of CPN0012.

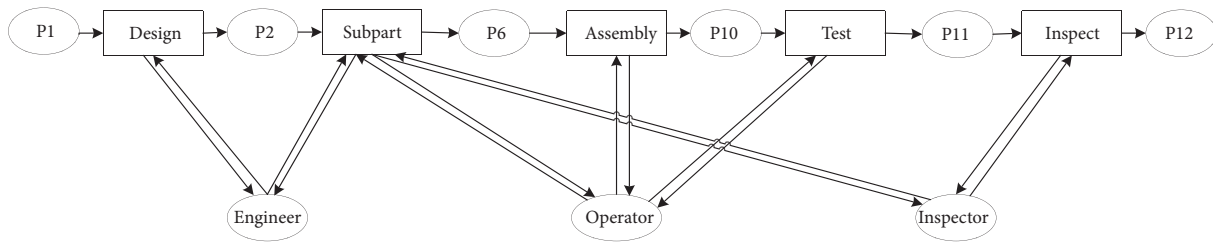


FIGURE 9: CPN model for the whole process.

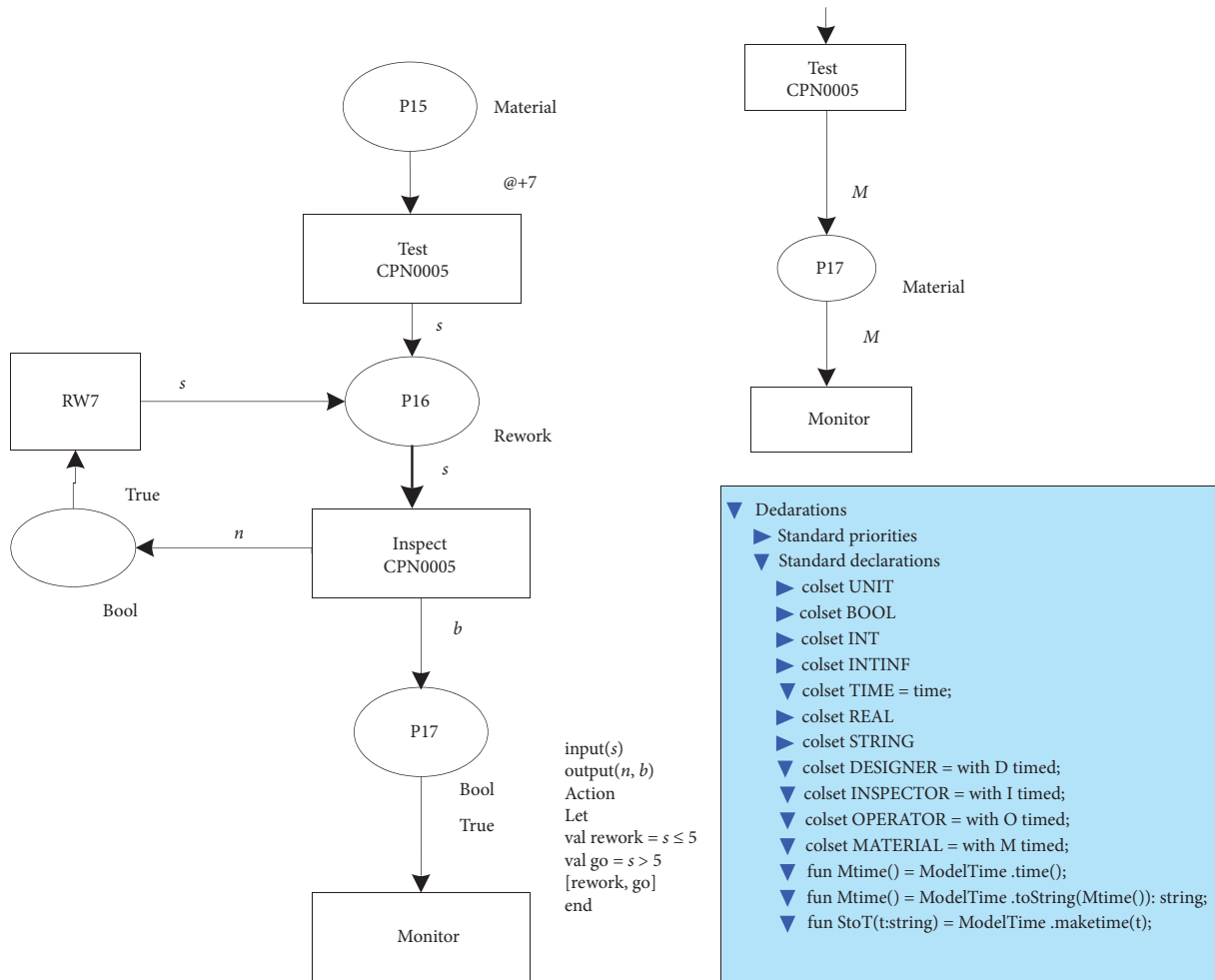


FIGURE 10: Rework structure and output transition.

top right-hand side of Figure 10. After CPN0005, the delivery unit is inspected and confirmed qualified; then, the token in the final place (indicating a finished delivery unit) will be consumed in another transition referred to as “monitor.” This monitor transition will consume the token and report the time stamp attached to the token, which is the total lead time.

In summary, we model the mainstream process as CPN and describe the rework structure, and the inspection has been modeled as a normal transition, which will drive the process forward. Furthermore, according to Definitions 1–6, the model meets all formal verification rules. The complete model is given in Figure 11.

5. Simulation Result and Analysis

5.1. Simulation Result. The data type has already been introduced in the previous section, and the model will be simulated 10,000 times to ensure all possible conditions could occur. The rework rate for each task before an inspection is 5%, which is one out of 20. These types of tasks appear three times at most in one work stream. Three tasks in the CPN0012 workflow stream need to be inspected. Hence, if the rework situation is uniformly distributed, the likelihood of it occurring three times in a row is $20 * 20 * 20 = 8000$. Simulating the CPN 10,000 times will theoretically generate the results for most possible conditions.

After 10,000 repeats of the simulation, the average lead time is 118.06 days. The best possible lead time is 109 days where resources are perfectly allocated and all tasks have been approved by inspection for just one time (see Tables 4 and 5 for detailed indicators). However, the deadline for this project is 100 days; therefore, it cannot meet the demands, and the process should be improved.

5.2. Result Analysis

5.2.1. Analysis of Critical Time. Based on the simulation results, we summarize the data on the start time as shown in Table 5. Before visualizing all the summed data, the actual earliest starting date (AESD), theoretical earliest starting date (TESD), and latest starting date (LSD) should be compared. The DELAY value is equal to AESD-TESD, indicating how much the actual start time is later than TESD, which may be due to resource constraints that are not theoretically earliest. The FAIL value is equal to AESD-LSD. When the FAIL value is greater than 0 (marked in red), it indicates that the actual start time of the task is greater than the latest start time of the task, which will inevitably cause the completion time of the entire system to exceed the specified time range. WINDOW is equal to LSE-TESD, which represents the buffer time of the task (i.e., the task starts on any day within the time window, and the entire manufacturing system will not exceed the specified completion time). If the DELAY value is greater than WINDOW, the manufacturing process cannot be completed on time, which is the same as the judgment using the FAIL value.

After comparing and listing all these data, we analyze them visually, as shown in Figure 12.

The last three tasks in the process have the same FAIL value because these three tasks are a simple one-stream process. Thus, it is the previous tasks that result in the positive FAIL value.

I0030 is an interesting task. The data show that I0033 has a FAIL value of 1, which means that this task can only begin one day after being scheduled. Its preceding resource-required task is S 0030. The data show that the FAIL value for S0030 is 0, which means that it is not S0030's fault that I0030 is late. Therefore, the only possible cause of the lateness of I0033 should be the limited Inspector resource.

I0012 is another failed task. The logic for analyzing task I0012 is the same as for I0033. The previous resource-required task has a FAIL value of -4 and the shortage of Inspector for task I0012 will, therefore, be solely responsible for the failure. Thus, critical issue one is caused by the shortage of inspectors.

Five of the tasks have no window at all. This is not acceptable, particularly when there are two tasks that might require rework. If any of the no-window tasks are reworked, the total service level will at least decrease to 95%, which is the service level for this task.

We have observed that most tasks have a delay time more than their windows. This could be due to many reasons. Consider the logic of the CPN; there are only three possible reasons for this: rework, limited resources, and bad workflow structure design. Yet, if we examine in further detail, tasks involving engineers generally have a delay time similar to or less than the window. Most of the delay problems occur for the inspectors. Thus, the most important resource shortage that causes this critical issue is also the shortage of inspectors.

5.2.2. Analysis about Reliability. To analyze reliability, we have calculated the following data, as shown in Table 6. We summarize the standard deviation (STD) and the average start time (AVG) of the simulation results, where FAIL is the difference between AVG and LSD, and $FAIL + 3 * STD$ is calculated based on the 6 sigma system.

Figure 13 lists a couple of critical issues. The 6 sigma system was introduced in the previous section. If the service level is set to 99.865%, the value of $FAIL + 3 * STD$ must be negative. Unfortunately, there are only four tasks that reach the standard.

We have observed that with the exception of I0034 and I0032, almost all inspection tasks have a relatively large standard deviation. Because inspection work is relatively short and predecessors of inspection tasks for different components may not end at the same time, the shortage of inspectors does not cause all inspection tasks to have a large standard deviation, and this is because of potential rework. I0032 has a much smaller standard deviation because the purchase process is rapid. I0034 is a similar case.

6. Improvement Strategy

Here, we will use the ECRS method and theory of constraints that was introduced in the Methodology section to propose improvements and suggest new scheduling strategies.

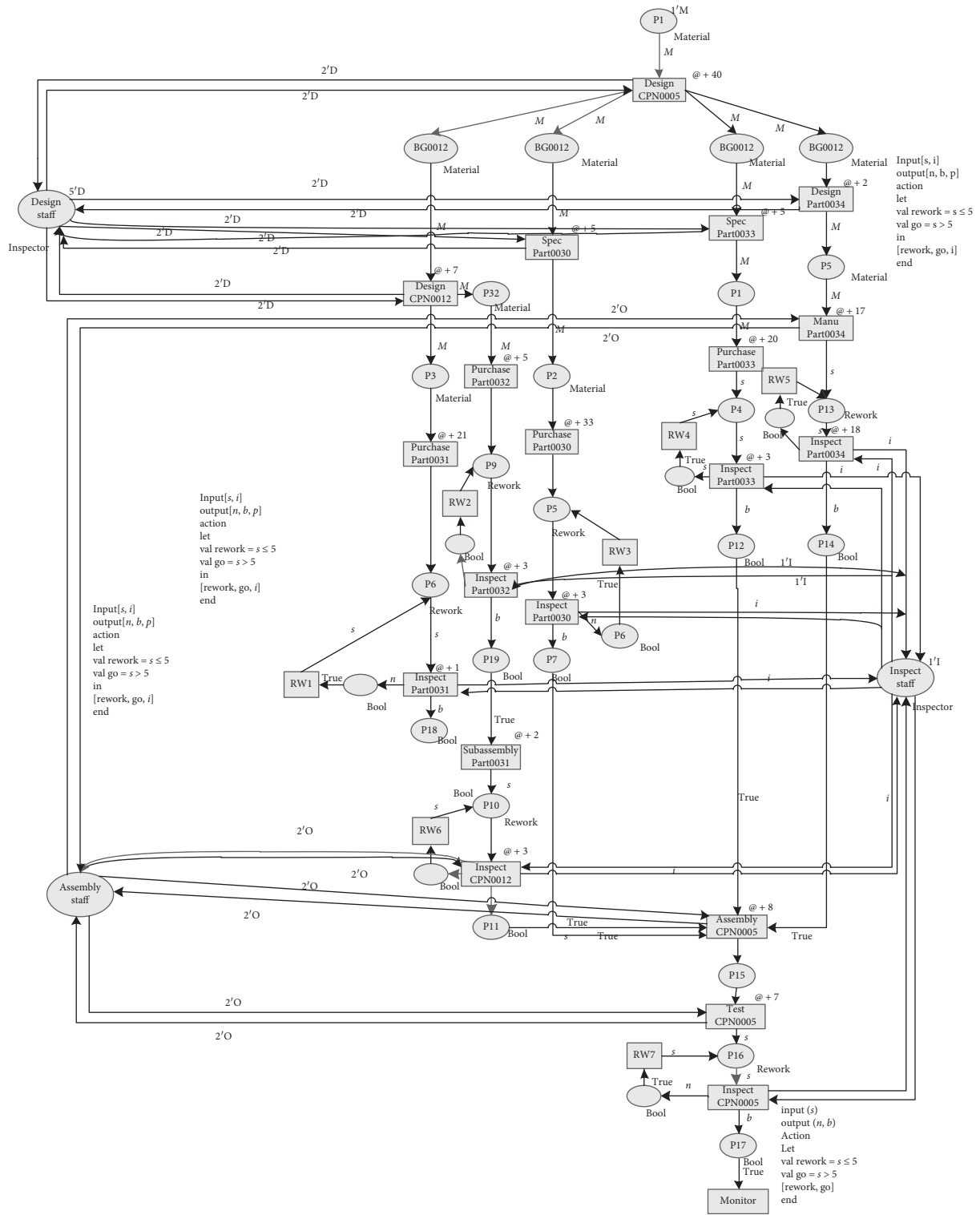


FIGURE 11: Original CPN model.

6.1. *Workflow Structure.* According to the theory of constraints, the critical path should first be identified, and then the entire system should be analyzed step by step. The stream of PART0030 is the critical path where all the tasks in it have no window at all. The stream of CPN0012 also has a relatively small window, which could be focused on after the

PART0030 stream. PART0033 and PART0034 have a relatively large window, which means that their priorities are in last place.

The workflow structure of PART0030 must be changed because the rework rate will directly lead to the total process reliability being less than 95%. This can be overcome

TABLE 5: Data summary.

Task	AESD	TESD	LSD	DELAY	FAIL	WINDOW
D0012	40	40	44	0	-4	4
D0034	40	40	57	0	-17	17
S0030	40	40	40	0	0	0
S0033	40	40	55	0	-15	15
A0005	87	78	78	9	9	0
A0012	69	69	73	0	-4	4
M0034	42	42	65	0	-23	23
T0005	95	86	86	9	9	0
I0005	102	93	93	9	9	0
I0012	81	71	75	10	6	4
I0030	77	75	75	2	2	0
I0031	68	68	72	0	-4	4
I0032	52	52	70	0	-18	18
I0033	65	65	75	0	-10	10
I0034	59	59	75	0	-16	16

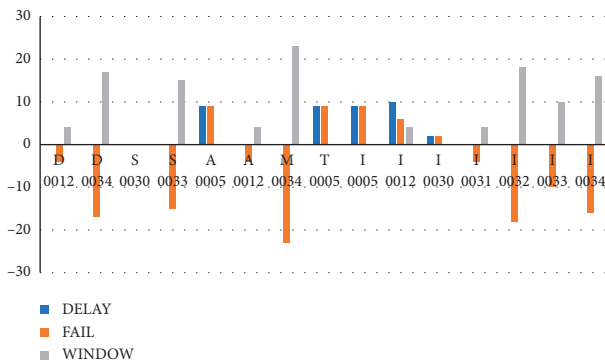


FIGURE 12: ESD comparison.

TABLE 6: Reliability data.

Task	AVG	LSD	FAIL	STD	6 sigma
D0012	42.01	44	-1.99	2.24	4.73
D0034	42.8	57	-14.2	2.89	-5.53
S0030	42.38	40	2.38	2.76	10.66
S0033	42.3	55	-12.7	2.75	-4.45
A0005	95.28	78	17.28	10.56	48.96
A0012	84.55	73	11.55	7.06	32.73
M0034	45.69	65	-19.31	5.03	-4.22
T0005	103.67	86	17.67	10.68	49.71
I0005	110.67	93	17.67	10.68	49.71
I0012	88.23	75	13.23	5.97	31.14
I0030	83.72	75	8.72	5.1	24.02
I0031	82.7	72	10.7	5.36	26.78
I0032	54.58	70	-15.42	3.54	-4.8
I0033	80.85	75	5.85	6.39	25.02
I0034	63.06	75	-11.94	5.12	3.42

through buffer times using the ECRS method, and one of which is to change the strategic objective in sourcing and, thus, alter the current suppliers. Suppliers who can deliver the product sooner are preferred. Several models, such as Kraljic’s model [37] and the strategic SRM model [38], can support this process. Yet, most of the time, a shorter lead time will lead to worse quality, which will increase the rework rate. Another way to reduce purchase time is to

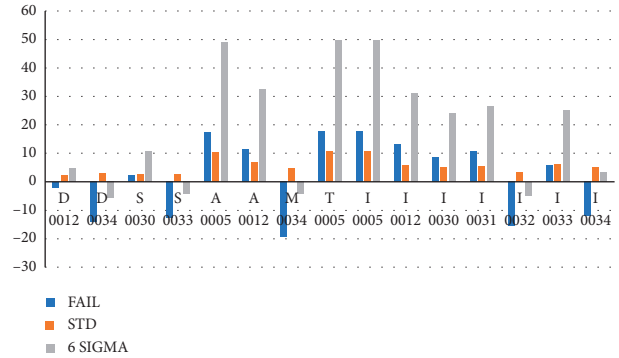


FIGURE 13: 6 sigma analysis.

provide technical support, such as via engineers or machines for the supplier. Studies have demonstrated that providing technical and resource support can improve the supplier performance in terms of quality and lead time [37]. Thus, engineers can be sent to the supplier to help develop and make PART0030. This approach can also eliminate the inspection task in that engineers that are assisting the purchasing process can help to ensure product quality.

6.2. *Assemble, Rework, and Inspection.* The appropriate strategy for eliminating rework has already been presented in the previous section. One of the best ways to mediate the negative impact of rework is to find a way to eliminate it. The best possible approach is to send engineers to suppliers to assist with the production process. This strategy has three theoretical positive impacts. The following decisions can be made based on the ECRS method.

One is that the incoming products do not need to be inspected anymore because engineers assisting the production process can do the job instead. The other positive impact is that there will not be any rework risk at all because the production is under inspection for the entire time.

An alternative is to have inspectors doing inspection along the assembly and test tasks. These two tasks might need more time for completion because of quality control, but the risk of large-scale rework is eliminated. Then, the task assembly and test will need a longer and more variable time for completion.

With the improvement strategy proposed, the new model is then simulated, and the execution time of each task and the number of various personnel is randomly generated. During this process, the 6 sigma system is still used as a neighborhood search rule to generate higher-quality candidate solutions to determine the manufacturing time and number of people for each task. Finally, the simulation results of the new manufacturing process are compared with the original model, and the results are as follows.

6.3. *Result Comparison.* According to the improvement suggestions, we have established a new scheduling scheme. The model is placed in Figure 14, and the data are summarized in Tables 7 and 8.

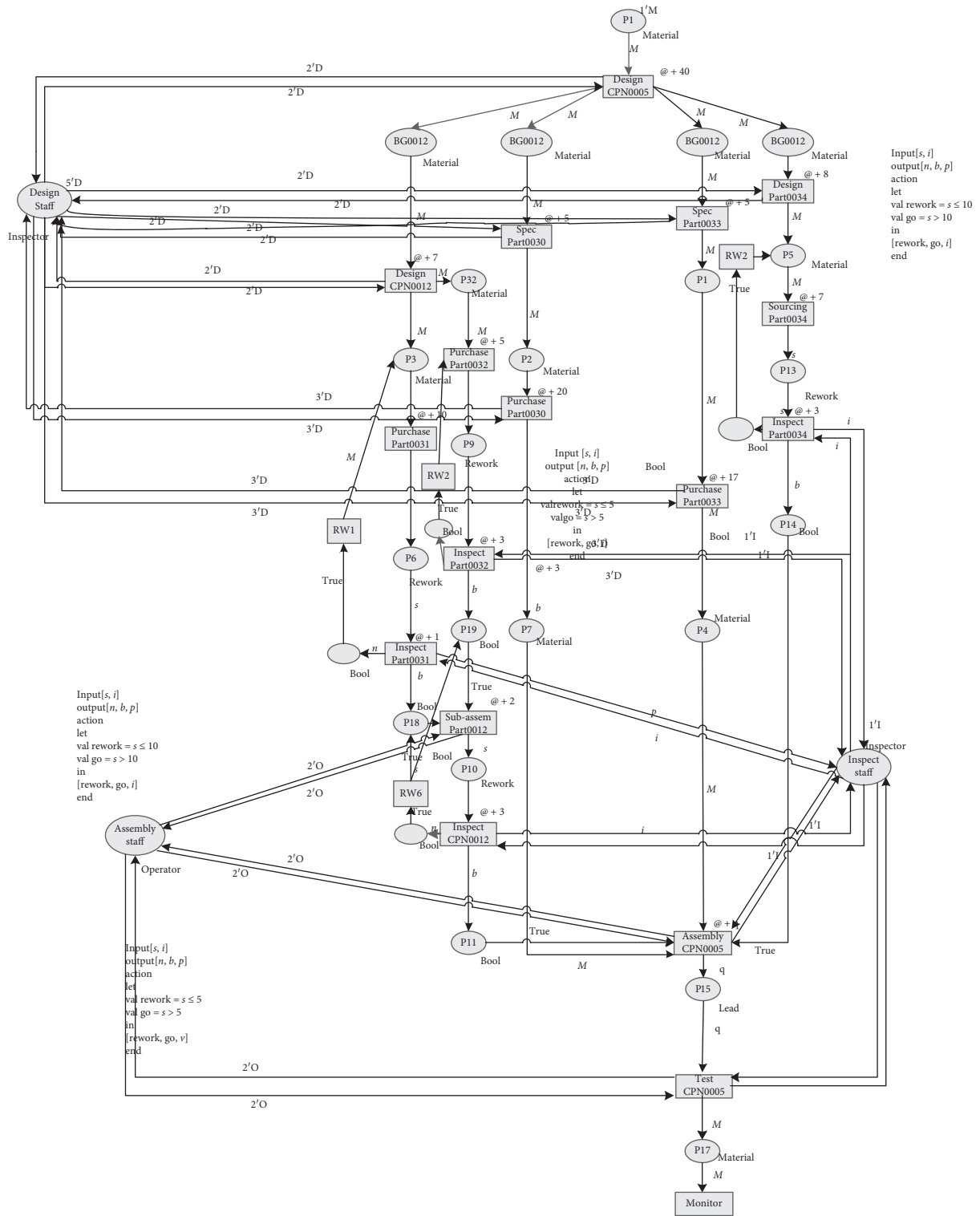


FIGURE 14: Optimized CPN model proposed.

TABLE 7: ESD comparison.

Task	AESD	TESD	LSD	DELAY	FAIL	WINDOW	ALSD	WDELAY
OTLT	109	100	100	9	9	0	202	102
NLT	79	76	100	3	-21	24	131	31

TABLE 8: Reliability comparison.

Task	AVG	LSD	FAIL	STD	6 sigma
OTLT	118.06	100	18.06	11.06	51.24
NTLT	86.7	100	-13.3	6.67	6.71

As shown in Table 7, the NTLT performs much better in every aspect. The actual earliest starting date improves by 38%. The DELAY value for NTLT is smaller than OTLT, which means that the process runs more efficiently due to the provision of sufficient resources. The FAIL value for NTLT is now negative and, thus, it is now possible to complete the task on time. NTLT also has a window, which means that it is not tightly scheduled; hence, there is spare time to cope with emergencies such as rework. The WDELAY for NTLT is substantially shorter than OTLT, and therefore, the worst possible conditions for the new workflow will perform much better than the old one.

As shown in Table 8, the average lead time is significantly reduced for the new workflow. The FAIL value now becomes negative and the process is, hence, more likely to succeed. There is also a slight reduction in standard deviation for the new workflow; thus, the new process does a better job in risk control. The 6 sigma value drops from over 50 to approximately 6. This is a demonstration of how reliable this process becomes comparing to the old workflow. The new workflow lies in the plus and minus two sigma range and there is, therefore, a 95% successful rate for the entire process.

7. Conclusion

This paper has presented an approach for analyzing and improving the workflow for the Phobos-Grunt SOPSYS delivery unit manufacturing process via a colored Petri net (CPN).

First, the gap of the current study in workflow optimization and CPN implementation was identified. There is no implication of CPN in the literature available for a space project workflow optimization. This paper filled this gap by presenting a comprehensive approach to analyze, optimize, and validate an existing time-sensitive, single unit, space instrument manufacturing workflow via a CPN. Second, this paper constructed a CPN model for the specific case. Some extended CPN theories were introduced to support the construction model. We have proposed a framework and method of analysis for the construction of the model. For example, a CPN structure for rework logic is developed to simulate the real case. Third, this research generated an improvement proposal according to the simulation data: the ECRS method and theory of constraints theories are used for generating the proposal.

The first practical implication of this work is the Phobos-Grunt SOPSYS delivery unit manufacturing process itself. The ultimate goal of this study is to reduce the manufacturing process lead time in the space project.

The results of the study have demonstrated its achievement. This research helps avoid the potential failure in the SOPSYS delivery unit manufacturing process and improves the reliability of it with a well-developed improvement strategy. The second implication of this study is its extended value for all the other normal time-sensitive single-unit manufacturing processes. Although the case modeling focuses on the Phobos-Grunt SOPSYS delivery unit manufacturing process, the logic of the process is similar to most other time-sensitive single-unit manufacturing processes. The same process and techniques can be used in other manufacturing systems for workflow optimization purposes in various space missions to the moon, Mars, and other planets.

Due to the relatively specific research background of this article, the data related to resources and time were obtained from the real SOPSYS unit manufacturing system. For this precise and customized instrument manufacturing process, accurate resource positioning must be achieved. Second, resource management systems often need to be manually modified for emergencies and resiliences. The resource management system could be built for a long-term, stable, and resource-intensive manufacturing system: the framework and techniques for building such a system are being further refined and developed using this case study's results as a basis. Overall, our research has provided some theory and experiences for workflow optimization in various space missions in future.

Data Availability

The data used in the article have been given in Table 3 of Section 4. The background information support this study can be found in "Polyu.edu.hk (2015) life on mars? award-winning polyu device could dig up the answer" (<https://www.polyu.edu.hk/openingminds/en/story.php?sid=3>).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors has been contributed equally to this work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant nos. 71772033, 71831003, and 71801031, in part by the Scientific and Research Funds of the Education Department of Liaoning Province of China under grant no. LN2019Q14, in part by the Natural Science Foundation of Liaoning Province of China (joint open fund for key scientific and technological innovation bases) under grant no. 2020-KF-11-11, and in part by the Department of Industrial and Systems Engineering of the Hong Kong Polytechnic University under grant no. H-ZG3K

References

- [1] T. M. Koulopoulos, *The Workflow Imperative*, Van Nostrand Reinhold, New York, NY, USA, 1995, https://www.researchgate.net/publication/229124468_The_Workflow_Imperative.
- [2] P. Lawrence, *Workflow Management Coalition, Workflow Handbook*, John Wiley & Sons, Hoboken, NJ, USA, 1997, <https://dl.acm.org/citation.cfm?id=272945>.
- [3] H. Hong-Jun and Y. Jue, "An airport pavement traffic simulation based on CPN," *International Journal of Engineering Business Management*, vol. 6, 2014.
- [4] W. Tan, W. Xu, F. Yang et al., "A framework for service enterprise workflow simulation with multi-agents cooperation," *Enterprise Information Systems*, vol. 7, pp. 523–542, 2013.
- [5] Y. Q. Lv, C. K. M. Lee, Z. Wu et al., "Priority-based distributed manufacturing process modeling via hierarchical timed color petri net," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 1836–1846, 2013.
- [6] J. W. Wang, W. H. Ip, R. R. Muddada et al., "On Petri net implementation of proactive resilient holistic supply chain networks," *The International Journal of Advanced Manufacturing Technology*, vol. 69, pp. 427–437, 2013.
- [7] R. Y. K. Fung, A. Y. M. Au, and A. W. H. Ip, "Petri Net-based workflow management systems for in-process control in a plastic processing plant," *Journal of Materials Processing Technology*, vol. 139, pp. 302–309, 2003.
- [8] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Upper Saddle River, NJ, USA, 1981.
- [9] K. Jensen, "Coloured petri nets and the invariant-method," *Theoretical Computer Science*, vol. 14, pp. 317–336, 1981.
- [10] K. Yung, C. W. Lam, S. M. Ko et al., "The Phobos-Grunt microgravity soil preparation system," *Acta Astronaut*, vol. 141, pp. 22–29, 2017.
- [11] K. V. Hee, O. Oanea, R. Post et al., "Yasper: a tool for workflow modeling and analysis," in *Proceeding of the Sixth International Conference on Application of Concurrency to System Design (ACSD'06)*, pp. 279–282, Turku, Finland, 2006.
- [12] J. Kurt, "A brief introduction to coloured petri nets," *Tools & Algorithms for Construction & Analysis of Systems*, Springer, Berlin, Heidelberg, 1997.
- [13] R. Dewan, A. Seidmann, and Z. Walter, "Workflow optimization through task redesign in business information processes," in *Proceeding of the Hawaii International Conference on System Sciences*, p. 240, IEEE, Kohala Coast, HI, USA, 1998.
- [14] R. Bartk, M. Jaska, L. Novk et al., "Workflow optimization with FlowOpt: on modelling, optimizing, visualizing, and analysing production workflows," in *Proceeding of the International Conference on Technologies and Applications of Artificial Intelligence*, pp. 167–172, IEEE, Chung-Li, Taiwan, 2011.
- [15] F. Crop, T. Lacornerie, X. Mirabel et al., "Workflow optimization for robotic stereotactic radiotherapy treatments: application of constant work in progress workflow," *Operations Research for Health Care*, vol. 6, pp. 18–22, 2015.
- [16] Y. Du, Z. Yu, B. Yang et al., "Modeling and simulation of time and value throughputs of data-aware workflow processes," *Journal of Intelligent Manufacturing*, vol. 30, no. 6, pp. 2355–2373, 2019.
- [17] B. Karlik, M. Uzam, M. Cinsdikici et al., "Neurovision-based logic control of an experimental manufacturing plant using neural net le-net5 and automation petri nets," *Journal of Intelligent Manufacturing*, vol. 16, no. 4-5, pp. 527–548, 2005.
- [18] C. Kasemset, P. Pinmanee, and P. Umarin, "Application of ECRS and simulation techniques in bottleneck identification and improvement: a paper package factory industry 4.0 for SMEs-EU horizon 2020 RISE program view project application of ECRS and simulation techniques in bottleneck identification and improvement: a paper package factory," in *Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference*, Hong Kong, China, December 2018, <https://www.researchgate.net/publication/276417414>.
- [19] M. À Piera, M. Narciso, A. Guasch et al., "Optimization of logistic and manufacturing systems through simulation: a colored petri net-based methodology," *Simulation*, vol. 80, pp. 121–129, 2004.
- [20] N. Wightkin and U. Buy, "Formal modeling of sequential function charts with time Petri nets," *IEEE Transactions on Control Systems Technology*, vol. 20, pp. 678–692, 2014.
- [21] W. M. Zuberek and W. Kubiak, "Timed petri nets in modeling and analysis of simple schedules for manufacturing cells," *Computers & Mathematics with Applications*, vol. 37, no. 11–12, pp. 191–206, 1999.
- [22] E. Villani and J. C. Pascal, "A petri net-based object-oriented approach for the modelling of hybrid productive systems," *Nonlinear Analysis*, vol. 62, no. 8, pp. 1394–1418, 2005.
- [23] M. Gao, "Modeling, service planning and service composition in knowledge-intensive collaborative workflows," *Ph.D. dissertation, School Management Science and Engineering*, Dongbei University of Finance and Economics, Dalian, China, 2013.
- [24] D. R. Kiran, "Six sigma," in *Total Quality Management*, Elsevier, Amsterdam, Netherlands, pp. 347–361, 2016.
- [25] D. Y. Mo, S. C. H. Ng, and D. Tai, "Revamping NetApp's service parts operations by process optimization," *INFORMS Journal on Applied Analytics*, vol. 49, no. 6, pp. 407–421, 2019.
- [26] V. Rafiee and J. Faiz, "Robust design of an outer Rotor Permanent magnet motor through six-sigma methodology using Response surface surrogate model," *IEEE Transactions on Magnetics*, vol. 55, no. 10, pp. 1–10, 2019.
- [27] L. G. Mkhaimer, M. Arafeh, and A. H. Sakhrieh, "Effective implementation of ISO 50001 energy management system: applying Lean six sigma approach," *International Journal of Business Management*, vol. 9, pp. 1–12, 2017.
- [28] Y. N. Al Khamisi, M. Khurshid Khan, and J. E. Munive-Hernandez, "Assessing quality management system at a tertiary hospital in Oman using a hybrid knowledge-based system," *International Journal of Engineering Business Management*, vol. 10, pp. 1–13, 2018.
- [29] F. A. A. Miranda, "Application of work sampling and ECRS (eliminate, combine, Re-lay out and simplify) principles of improvement at TO1 Assembly," in *Proceedings fo the 21stASEMEP National Technical Symposium*, Las Vegas, NV, USA, 2018, <http://www.onsemi.com/site/pdf/SANYO-ApplicationofWorkSamplingandECRS-FINAL2.pdf>.
- [30] T. Kis, D. Kiritsis, P. Xirouchakis et al., "A petri net model for integrated process and job shop production planning," *Journal of Intelligent Manufacturing*, vol. 11, no. 2, pp. 191–207, 2000.
- [31] W. M. Zuberek and W. Kubiak, "Timed petri nets in modeling and analysis of simple schedules for manufacturing cells," *Computers & Mathematics with Applications*, vol. 37, no. 11–12, pp. 191–206, 1999.
- [32] X. You, L. Zhang, and J. Jiao, "Supply chain configuration modeling based on colored Petri-nets," in *Proceedings of the*

- IEEE International Conference on Management of Innovation and Technology*, pp. 921–925, IEEE, Kauia, HI, USA, 2006.
- [33] M. Y. Marov, V. S. Avduevsky, E. L. Akim et al., “Phobos-Grunt: Russian sample return mission,” *Advances in Space Research*, vol. 33, pp. 2276–2280, 2004.
- [34] Phobos-Grunt’s sad return,” *Astronomy & Geophysics*, vol. 53, pp. 10–11, 2012.
- [35] K. Jensen, L. M. Kristensen, and L. Wells, “Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems,” *International Journal on Software Tools for Technology Transfer*, vol. 9, pp. 213–254, 2007.
- [36] CPN Tools, 2018, <http://cpntools.org/>.
- [37] Monitors-CPN Tools, 2018, <http://cpntools.org/2018/01/12/monitors/>.
- [38] W. P. Oosterhuis and M. J. Severens, “Performance specifications and six sigma theory: clinical chemistry and industry compared,” *Clinical Biochemistry*, vol. 57, pp. 12–17, 2018.