



Article

An Industrial Case Study on the Monitoring and Maintenance Service System for a Robot-Driven Polishing Service System under Industry 4.0 Contexts

Yuqian Yang ¹, Maolin Yang ¹, Siwei Shangguan ¹, Yifan Cao ¹, Wei Yue ¹, Kaiqiang Cheng ²
and Pingyu Jiang ^{1,*}

¹ State Key Laboratory of Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710054, China

² Shandong Cloud Imagination Technology Co., Ltd., Yantai 264000, China

* Correspondence: pjiang@mail.xjtu.edu.cn

Abstract: Remote monitoring and maintenance are important for improving the performance of production systems. However, existing studies on this topic usually focus on the monitoring and maintenance of the working conditions of the equipment and pay relatively less attention to the processing craft and processing quality. In addition, as far as we know, there are relatively few industrial case studies on the real applications of remote monitoring and maintenance systems that include both conventional and advanced maintenance techniques under the context of Industry 4.0. Addressing these issues, an industrial case study on the monitoring and maintenance service system for a robot-driven carbon block polishing service system is presented, including its application background and engineering problems, software/hardware architecture and running logic, the monitoring and maintenance-related enabling techniques, and the configuration and operation workflows of the system in the form of screenshots of the functional WebAPPs of the software system. The case study can provide real examples and references for the industrial application of remote monitoring and maintenance service systems on industrial product service systems under the context of Industry 4.0. Advanced techniques such as the Industrial Internet of Things, digital twins, deep learning, and edge/cloud/fog computing have been applied to the system.

Keywords: Industry 4.0; remote monitoring; maintenance; deep learning; digital twin



Citation: Yang, Y.; Yang, M.; Shangguan, S.; Cao, Y.; Yue, W.; Cheng, K.; Jiang, P. An Industrial Case Study on the Monitoring and Maintenance Service System for a Robot-Driven Polishing Service System under Industry 4.0 Contexts. *Systems* **2023**, *11*, 376. <https://doi.org/10.3390/systems11070376>

Academic Editor: Paolo Visconti

Received: 28 June 2023

Revised: 16 July 2023

Accepted: 21 July 2023

Published: 22 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background and Engineering Problems

Anode carbon blocks (Figure 1a) are important consumables during aluminum production, and the production of anode carbon blocks mainly includes the processes of mixing, kneading, forming, roasting, and polishing. At present, the polishing process is commonly conducted manually in China, and this leads to issues such as environmental pollution, occupational diseases among workers (e.g., pneumoconiosis), high labor costs, and recruiting difficulty [1]. Under such a background, YX company developed a kind of robot-based processing line for carbon block polishing to replace the conventional manual polishing process, as shown in Figure 1b,c. However, the processing line was a newly developed equipment and its reliability still require further proving. Therefore, the customer companies were hesitant to buy the processing line. In this regard, YX developed a kind of industrial product service system (IPSS) based on the processing line and sells polishing services rather than the processing line itself [2]. However, the reliability of the processing line is still an issue of concern, and YX must develop a kind of monitoring and maintenance service system for the polishing processing line-based IPSS.

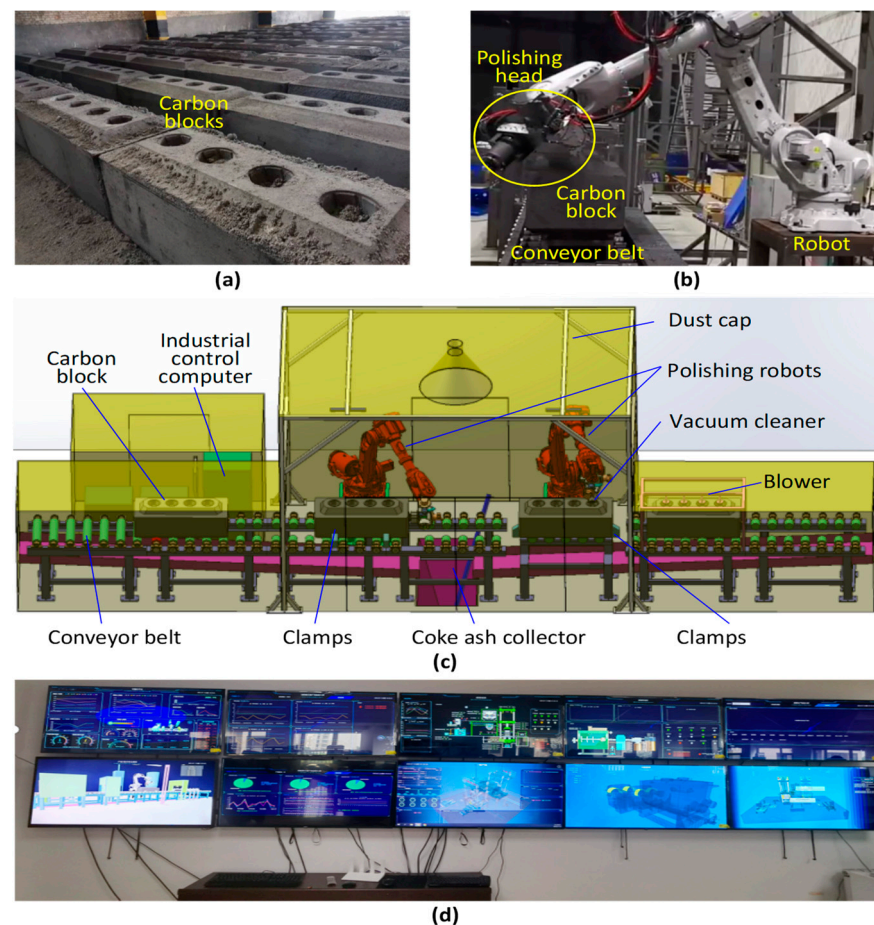


Figure 1. Robot-based carbon block polishing processing line and its control room displaying the remote monitoring and maintenance system: (a) carbon blocks, (b) real photo of robot-based polishing processing line, (c) illustration of entire polishing processing line, (d) monitoring screens of the processing line in the control room.

On the other hand, advanced techniques in Industry 4.0 contexts, such as the Industrial Internet of Things [3,4], cyber-physical systems [5], digital twins [6], deep learning [7], and edge/cloud/fog computing [8], have boosted the fast development of intelligent and remote monitoring/maintenance systems together with the smart IPSS [9].

However, as far as we know, currently there is a lack of examples of the above-mentioned systems in real industrial applications, especially when the purpose is to maintain equipment working conditions/processing craft/processing quality through remote monitoring and maintenance at the same time. Moreover, there are few examples of integration using conventional and advanced maintenance techniques under the context of Industry 4.0 at the same time.

1.2. Related Works

Under the context of Industry 4.0, using advanced information and computer science techniques for intelligent monitoring and maintenance of production systems has become an important research topic [10]. Much research has been devoted to the establishment of IPSSs and their corresponding maintenance and monitoring systems. For example, from the perspective of remote monitoring of manufacturing production lines, Chen [11] established a kind of intelligent production line monitoring architecture enabled by a wireless sensor network and RFID, and the architecture was able to realize real-time multi-threaded production data collection/storage and workpiece tracking monitoring for discrete manufacturing production lines. Li et al. [12] discussed the strategy of integrating production techniques with advanced information technology such as artificial intelligence,

edge computing, digital twins, etc. for intelligent monitoring and maintenance. Mofidul et al. [13] developed a remote monitoring system for an automatic packaging production line based on a programmable logic controller (PLC), and the system can replace existing industrial SCADA systems with ordinary sensors and control apps based on mobile applications, thus reducing the costs of building the monitoring systems. Mourtzis et al. [14] developed a conditional preventive maintenance system that can suggest maintenance strategies according to shop-floor data. From the perspective of production line maintenance under the context of Industry 4.0, Cheng et al. [15] developed an edge–cloud collaborative maintenance architecture for bearing production lines. Following a roadmap of “edge-side sensor data acquisition → edge-side data cleansing → cloud-side data analyzing,” the architecture can realize a reduction in maintenance costs and improve the equipment utilization rate by predicting equipment failure. Ayvaz and Alpay [16] proposed a data-driven predictive maintenance system for production lines. The system can detect potential production line faults by analyzing the real-time production data collected from sensors using a random forest ensemble learning framework. Cachada et al. [17] developed an architecture of intelligent and predictive maintenance framework under the context of Industry 4.0. Enabled by the Internet of Things, deep learning, etc., the framework can support equipment/production line fault prediction and monitor management, condition-based maintenance, etc. Zhang et al. [18] developed an Internet of Things-driven smart mine maintenance system, and the system is characterized by its integration of real-time acquisition, event-driven operation rules building, and maintenance knowledge fusion. From the perspective of IPSS, Qi and Tao [19] established a kind of industrial product service system based on cloud/fog/edge computing, and the system can provide enhanced product service management and control based on high-speed and low-delay collection and analysis of multi-level real-time production data of production lines. Zheng et al. [20] conducted a review of smart product service systems and discussed methods to improve the smartness of industrial service activities through the application of advanced intelligence and information techniques. Shihundla et al. [21] developed an integrated product service system from an equipment lifecycle perspective. Supported by the data collected by sensor networks and edge–cloud collaborated computing architecture, the system can realize data-driven equipment health assessment and product service management.

All the works above can provide references for the development of monitoring and maintenance systems for robot-driven smart product service systems. However, how exactly to effectively integrate the techniques of the Internet of Things, deep learning, etc. for the smart monitoring and maintenance of robot-driven processing lines still requires detailed exploration. How to implement a service system based on the processing line for the benefit of both the service provider and consumer companies in real commercial scenarios still requires further testing and discussion.

In this regard, YX developed a remote monitoring and maintenance service system for the polishing processing line-based IPSS. The monitoring and maintenance service system can provide customized monitoring and maintenance service through web-based software based on the processing line hardware. Conventional monitoring and maintenance techniques together with some advanced techniques under the context of Industry 4.0 have been applied in the system. The monitoring screens (deployed in the control room of the processing line in the customer company) displaying monitoring contents on working conditions/processing craft/polishing quality are illustrated in Figure 1d.

On this basis, this paper provides a case study on the design, development, and deployment of the monitoring and maintenance service system, and it aims at providing examples of the industrial applications of the above-mentioned systems and techniques.

The rest of the paper is organized as follows. Section 2 introduces the architecture and overall run-time logic of the hardware/software of the monitoring and maintenance service system together with the monitoring and maintenance domain technologies that support the realization of the system. Section 3 introduces how the system was deployed in

a customer company using combinations of screenshots of the functional WebAPPs of the system. Section 4 concludes the works and discusses the contributions and future works.

2. Systematic Architecture and Key Enabling Technologies

In this section, the software/hardware architecture and the key enabling techniques of the monitoring and maintenance service system of the polishing processing line-based IPSS are demonstrated.

2.1. System Architecture and Working Logic

2.1.1. System Architecture

A browser/server and configuration/operation separated architecture is applied for a monitoring and maintenance service system to enable convenient configuration of the monitoring and maintenance service according to specific customer requirements, as illustrated in Figure 2. The configuration system was deployed in the provider company (i.e., YX company), and it is mainly for generating a configuration file that defines the services provided by the provider company to specific customers. The operation system was deployed in the customer company, and it has to be activated by the configuration file before providing the monitoring and maintenance services. The entire architecture can be roughly separated into four layers, as listed below. It is worth mentioning that under the contexts of Industry 4.0 and Industrial IoT, layered architectures are usually applied to organize the components of the systems, and a four-layer structure is commonly applied, with the first layer for perception and the fourth layer for application [22–24].

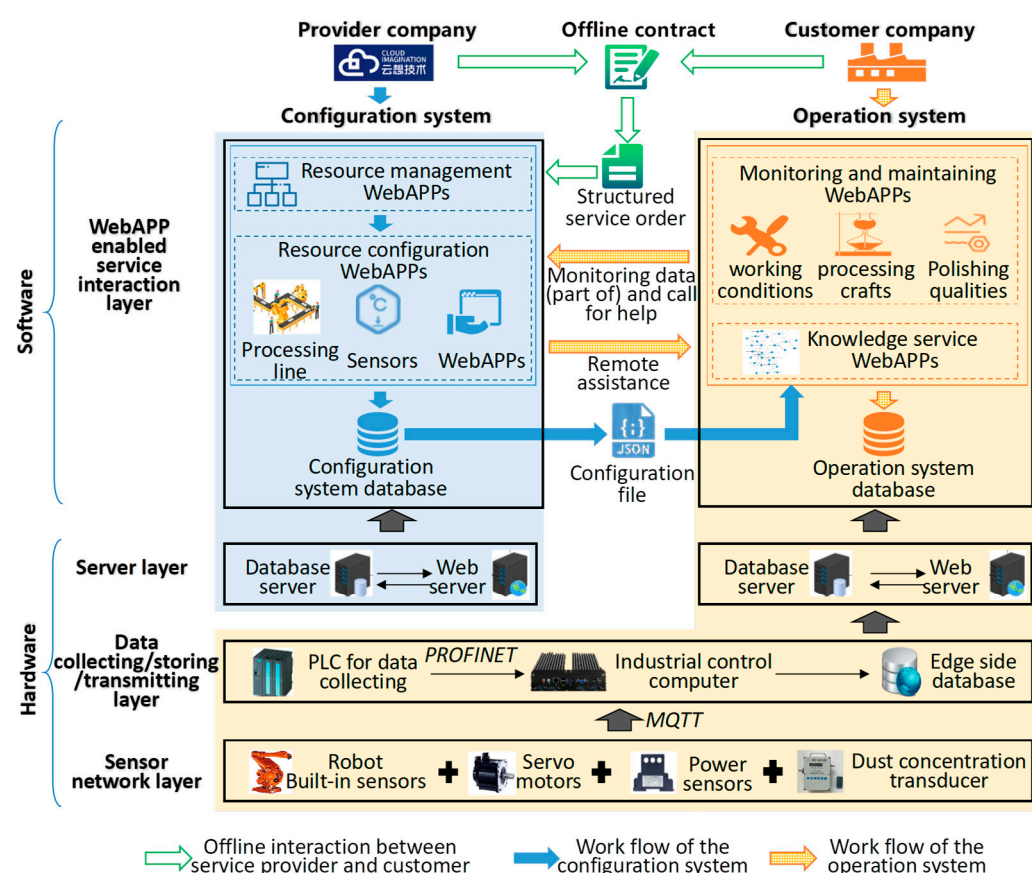


Figure 2. The architecture and working logic of the monitoring and maintenance system.

- **Sensor network layer:** This layer is mainly for collecting the edge-side operation data of the robot-based polishing processing line, and the sensors can be mainly separated into two categories. The first category is the built-in sensors of the processing line.

This mainly includes the built-in sensors of ABB-6700 (i.e., the robot applied in the case) and the servo motors in the polishing head. The signals monitored by the built-in sensors in this case study include the torques, the rotation angles of the axes of the robot, the position coordinates of the end effector of the robot (i.e., the polishing head), and the currents and torques of the polishing head. The second category is additionally mounted sensors, which mainly include three power sensors for the entire processing line/the robot/the polishing head, and a dust concentration transducer in the dust cover of the processing line. It is worth mentioning that message queuing telemetry transport (MQTT) protocol was applied to support the data transmission among the equipment in the sensor network layer and the data collecting/storing/transmitting layer due to its reliability in low bandwidth and unstable network environments, which was the case of the working environment of the carbon block polishing processing line.

- Data collecting/storing/transmitting layer (i.e., lower computer layer): This layer mainly contains PLC for deploying the programs for collecting the raw data from the sensors, an industrial computer for deploying the programs for preprocessing the raw data into flow data for high-frequency transmitting, and an edge-side database for temporarily storing the data, and a process field network (PROFINET) protocol was applied to support the data transmission between the PLC and the industrial control computer.
- Server layer (i.e., upper computer layer): The previous two layers are both deployed in the customer company, whereas the server layer is deployed in both the provider and the customer company. More specifically, both the provider and the customer deployed a web server for running the software of the configuration and the operation system and a database server to store all of the data generated during the service configuration and operation process.
- WebAPP-enabled service interaction layer: This layer is supported by the software of the configuration and operation systems, and the functions of the software were developed in the form of WebAPPs (i.e., web-enabled apps), each of which has relatively independent but related functions (as introduced in Section 3, where each bullet point indicates a WebAPP). In total, 12 functional WebAPPs were developed for the configuration system, and they can be roughly separated into resource management WebAPPs and resource configuration WebAPPs. Another 24 functional WebAPPs were developed for the operation system, and they can be roughly separated into monitoring and maintenance WebAPPs on working conditions/processing craft/polishing quality and knowledge service WebAPPs. The details of the WebAPPs are introduced in Section 3.

2.1.2. Working Logic

The working logic during the service interaction between the provider and the customer companies is illustrated with three different types of arrows in the WebAPP-enabled service interaction layer in Figure 2, and the interaction can be roughly separated into three stages, as listed below.

- Generating service orders through offline interaction: This stage indicates the offline interaction between the provider company and the customer company, and the result is a contract that includes a structured service order (examples of the structured service order can be found in our previous work [2]).
- Generating configuration files through the configuration system: As mentioned in Section 2.1.1, the configuration system contains resource management WebAPPs and resource configuration WebAPPs. Firstly, the managers from the provider company defines and manages the service resource that the provider company has using the resource management WebAPPs. The resource to be managed includes the configuration modules of the robot-based polishing processing line, different types of configurable sensors, and the monitoring/maintenance service-related functional WebAPPs in the

operation system. Secondly, the managers from the provider company define specific configuration files for specific customer companies according to the service orders generated in the previous stage. The configuration file determines which configurable modules are contained in the processing line for a specific customer company, which sensors are mounted, and which monitoring/maintenance service-related WebAPPs are granted. The configuration files were generated in encrypted JSON files.

- Conducting monitoring and maintenance service through the operation system: After being activated with the encrypted configuration file generated from the configuration system, the operation system can be used to support monitoring and maintenance services for the polishing processing line deployed in the customer company. During the process, if the customer companies are not satisfied with the service provided by the WebAPPs, they can send part of the monitoring data to the supplier company and call for help. Correspondingly, the provider can analyze the data and provide remote assistance if necessary.

2.2. Key Enabling Techniques

In total, 14 key enabling techniques were applied in the monitoring and maintenance service system, and all of these techniques were contained in the functional WebAPPs in the operation system. The classifications and the interactions among the techniques are illustrated in Figure 3.

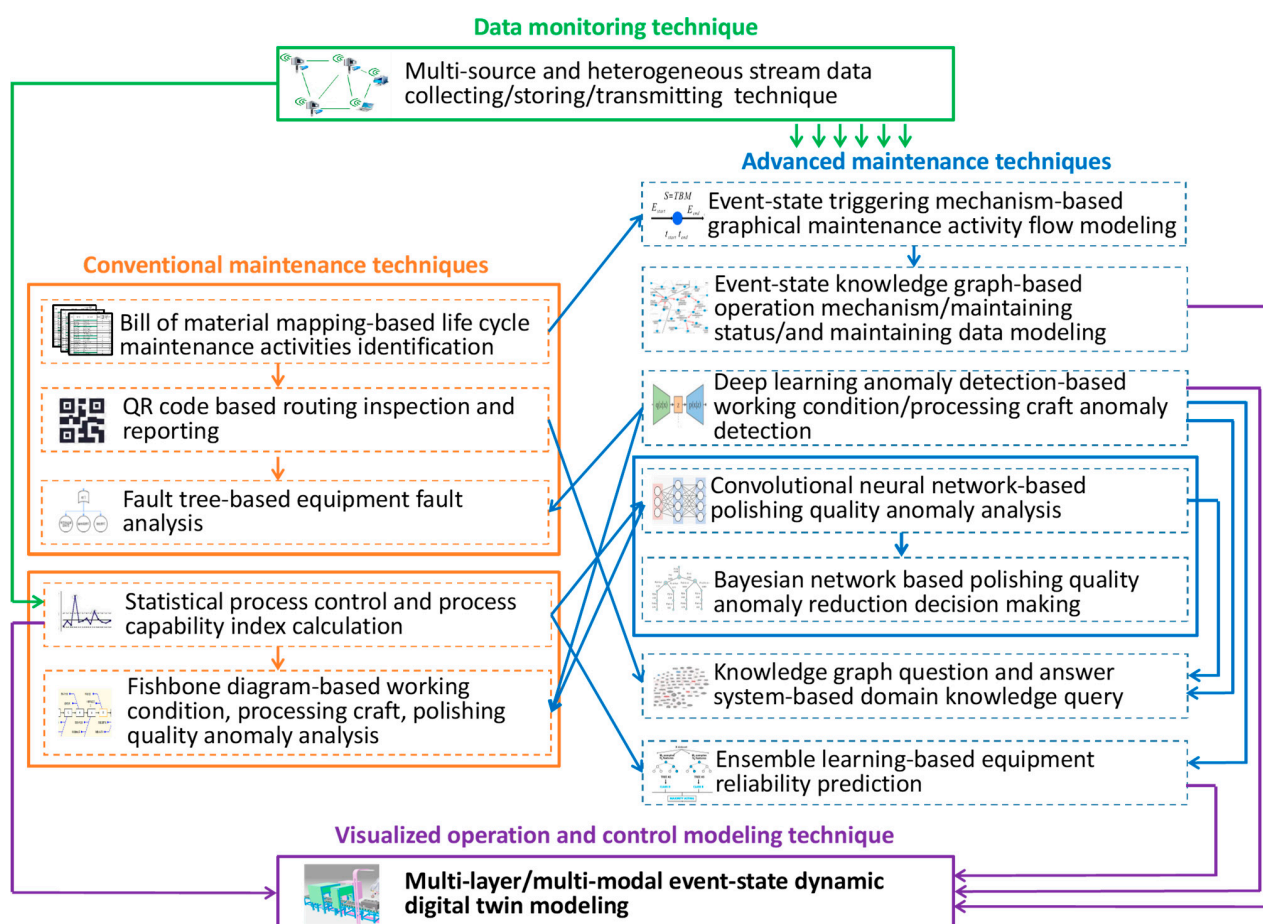


Figure 3. The key enabling techniques applied in the monitoring and maintenance system.

2.2.1. Data Monitoring

Multi-source and heterogeneous stream data collecting/storing/transmitting technique: This technique is for providing the required data for the conventional and advanced maintenance techniques and the visualized operation and control modeling technique.

2.2.2. Conventional Maintenance Techniques

- Bill of material (BOM) mapping-based life cycle maintenance activity identification: This technique is for identifying the essential maintenance activities of the processing line through the mapping between design BOM and maintenance BOM.
- QR code-based routing inspection and reporting: Routing inspection plans of the processing line can be established according to the graphical maintenance activity flow model (in advanced techniques), and the QR code-based inspection reporting technique can guarantee that the inspectors complete the inspection reports at designated times and locations.
- Fault tree-based equipment fault analysis: If any fault or anomaly situations are detected by the routing inspection or other anomaly detection techniques, this technique can help the engineers analyze the reasons that caused the fault or the anomaly situations. In addition, the technique can help store the analysis results in newly established fault trees as domain knowledge for further reuse (further introduced in the deep learning anomaly detection-based working condition/processing craft anomaly detection technique in Section 2.2.3).
- Statistical process control and process capability index calculation: This technique is for analyzing whether the processing line works properly (i.e., whether the carbon blocks were polished properly) through statistical calculations and provides suggestions on how to improve the process capability of the processing line.
- Fishbone chart-based working condition, processing craft, polishing quality anomaly analysis: This technique, driven by deep learning anomaly detection algorithms, serves as a supporting tool for analyzing any fault or anomaly situations. This technique can be also helpful for domain knowledge accumulation and reuse.

2.2.3. Advanced Maintenance Techniques

- Event-state triggering mechanism-based graphical maintenance activity flow modeling [25]: This technique can build the graphical maintenance activity flow model of a processing line, and the graphical model can be visually read by human engineers and implemented with computer programs at the same time. The modeling methods can be described in three parts, as illustrated in Figure 4.

The first part is to define the meta-model of the graphical maintenance activity flow modeling method. Four types of maintenance activity blocks (MABs) are defined, including time-based maintenance (TBM), which indicates the maintenance activities arranged by time periods; inspection-driven condition-based maintenance (ICBM), which indicates the repairing or maintenance activities performed when anomalies are detected during routine inspection; data-driven condition-based maintenance (DCBM), which indicates the repairing or maintenance activities performed when anomalies are detected by monitoring system; and breakdown maintenance (BM), which indicates the repairing or maintenance when the processing line suddenly breaks down. Furthermore, TBM can be separated into three statuses, including *To be completed*, which means that the time for the maintenance activity has not come yet; *Completed*, which means that the time has come and the maintenance activity has been completed; and *Missed*, which means that the time has come but the activity had not been completed. Each of the maintenance activities above can be described as

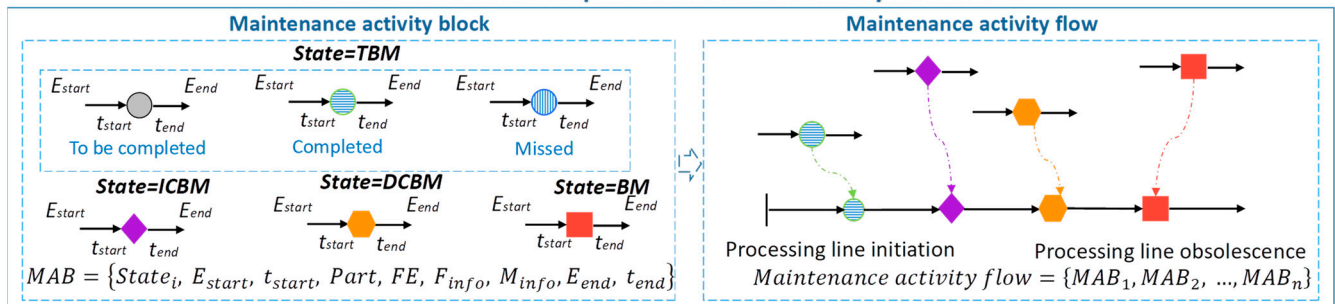
$$MAB_i = \{State_i, E_start, t_start, Part, FE, F_info, M_info, E_end, t_end\} \quad (1)$$

where MAB_i indicates the i th MAB for a processing line; $State_i$ indicates whether the MAB is a TBM, an ICBM, a DCBM, or a BM; E_start indicates an event that triggers the MAB; t_start indicates the start time of the MAB; $Part$ indicates the target part to be maintained or repaired; FE indicates the field engineer who is responsible for the MAB; F_info indicates the fault or anomaly information during the MAB; M_info indicates the detailed maintenance

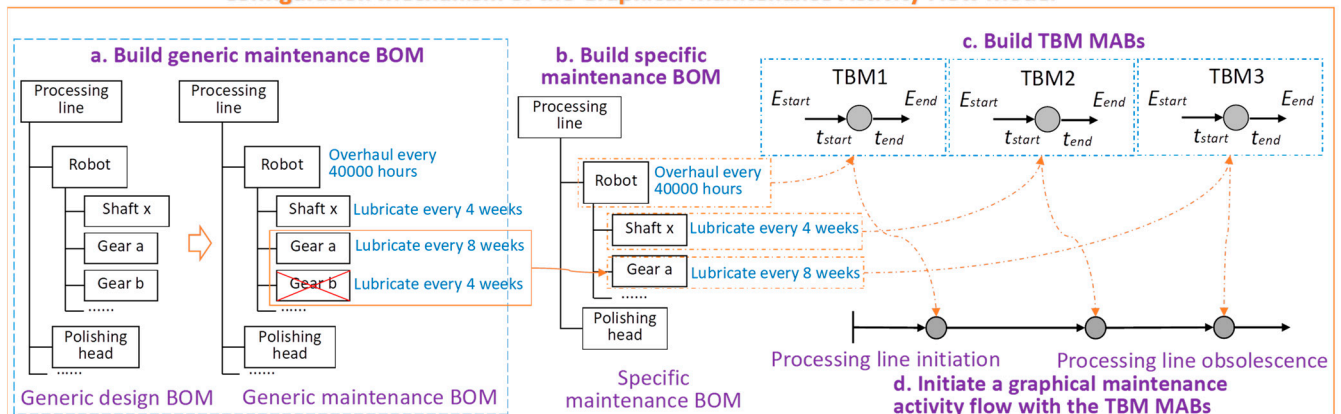
records of the MAB; E_{end} indicates an event that indicates the completion of the MAB; and t_{end} indicates the time when the MAB is completed.

The second part is the configuration mechanism of the graphical maintenance activity flow model. During this process, the BOM mapping-based life cycle maintenance activity identification technique in Section 2.2.2 is used. Firstly, the generic design BOM of the processing line is defined, based on which the generic maintenance BOM of the processing line can be defined correspondingly. Secondly, by determining the specific design BOM of a processing line the specific maintenance BOM can be determined correspondingly. Thirdly, each of the maintenance activities in the specific maintenance BOM is abstracted into a TBM MAB. Fourthly, the TBM MABs are serialized according to their cycling time periods to initiate a graphical maintenance activity flow of the specific processing line.

The Meta Model of the Graphical Maintenance Activity Flow Model



Configuration Mechanism of the Graphical Maintenance Activity Flow Model



Operating Mechanism of the Graphical Maintenance Activity Flow Model

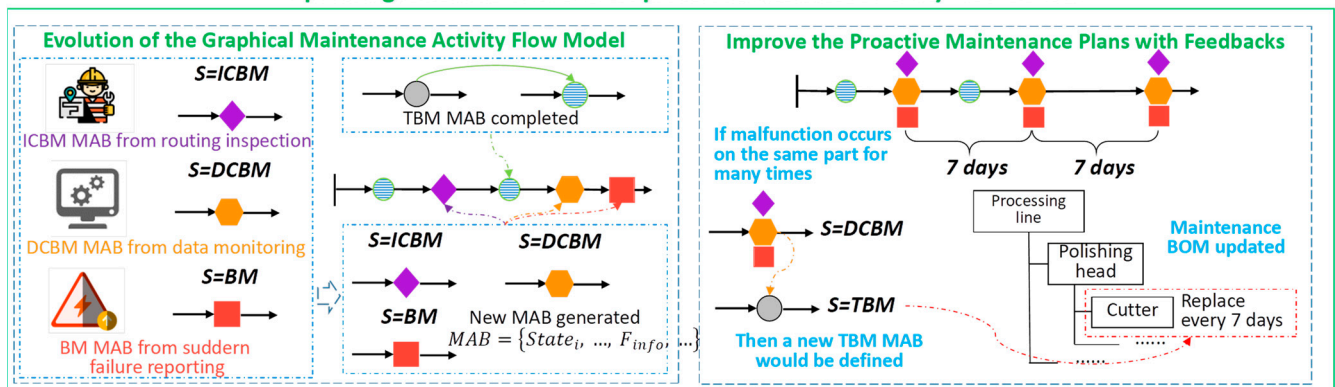


Figure 4. The technique for event-state triggering mechanism-based graphical maintenance activity flow modeling.

Table 1. The training steps of the deep learning-based anomaly detection model (using transformer-VAE [26] as an example).

| Algorithm Transformer-VAE | |
|---------------------------|--|
| Input: | Training set $X = \{x_t\}_{t=1}^{T_x}$ // x_t indicates a piece of training datum |
| Initialization: | Random initialized θ_0, Φ_0 // θ_0 , and Φ_0 are the predefined parameters of the encoder and decoder |
| Output: | Transformer-VAE parameters θ^*, Φ^* // θ^* , and Φ^* are the parameters of the encoder and decoder after training |
| 1: Repeat | |
| 2: | Sample x_t in the minibatch |
| 3: | Encoder: $\mu_{zt} \leftarrow f_\Phi(x_t)$ // $f_\Phi(x_t)$ is the encoder, μ_{zt} is a posterior probability distribution function of latent vector z_t |
| 4: | Sampling: $z_t \leftarrow \mu_{zt} + \epsilon \odot \sigma_{z_t}$, $\epsilon \sim \mathcal{N}(0, I)$ // z_t indicates the latent vector, ϵ is a vector generated from standard Gaussian distribution |
| 5: | Decoder: $\mu_{xt} \leftarrow f_\theta(z_t)$ // $f_\theta(z_t)$ is the decoder, μ_{xt} indicates the vector after been reconstructed |
| 6: | Compute reconstruction loss: $\mathcal{L}_{\text{rec}} = -\log p_\theta(x_t z_t) = -\log N(x_t; \mu_{xt}, \sigma_x^2 I)$ |
| 7: | Back-propagate the gradients |
| 8: | Until minimum loss is reached. |

The third part is the operating mechanism of the graphical maintenance activity flow model, and it is mainly for updating the initial graphical maintenance activity flow during the operation of the processing line. Firstly, the TBM MABs with the status of *To be completed* are updated to *Completed* or *Missed* according to the real maintenance situations. Secondly, ICBM MABs are added when anomalies are detected and corresponding maintenance activities are conducted during routine inspection, DCBM MABs are added when anomalies are detected by the monitoring system and corresponding maintenance activities are conducted, and BM MABs are added if a sudden breakdown occurred and corresponding maintenance activities are conducted. In addition, after collecting enough records of ICBM/DCBM/BM MABs, some of these MABs can be changed to TBM MABs, and the maintenance BOM is updated correspondingly.

- Event-state knowledge graph-based operation mechanism/maintenance status/and maintenance data modeling: This technique is for representing the operation mechanism, maintenance activity flow, working conditions, maintenance status, and historical monitoring data in the form of an event-state knowledge graph. This way, all of this information can be visually readable to human engineers and can be directly coded with computer programs at the same time. The implementation of this technique can be found in our previous work [2].
- Deep learning anomaly detection-based working condition/processing craft anomaly detection: As illustrated in Figure 5, using the working condition data (e.g., the current of the polishing head motors) or processing craft monitoring data (e.g., the working pressure of each polishing step) when the processing line works fine as training data, deep learning-based anomaly detection models can be trained. The trained anomaly detection models first encodes the input training data and then decode them into reconstructed data, and the similarity between the input and the reconstructed data would be high enough (e.g., $f_{\text{similarity}}(x) > \eta$). The pseudo-code of the training steps is demonstrated in Table 1. After being trained, the anomaly detection model can detect any type of abnormal time-series data of working conditions/processing craft, and thus it can provide a pre-alarm for more serious malfunctions. On this basis, the fault tree-based technique in Section 2.2.2 can be used to further analyze an abnormal situation.

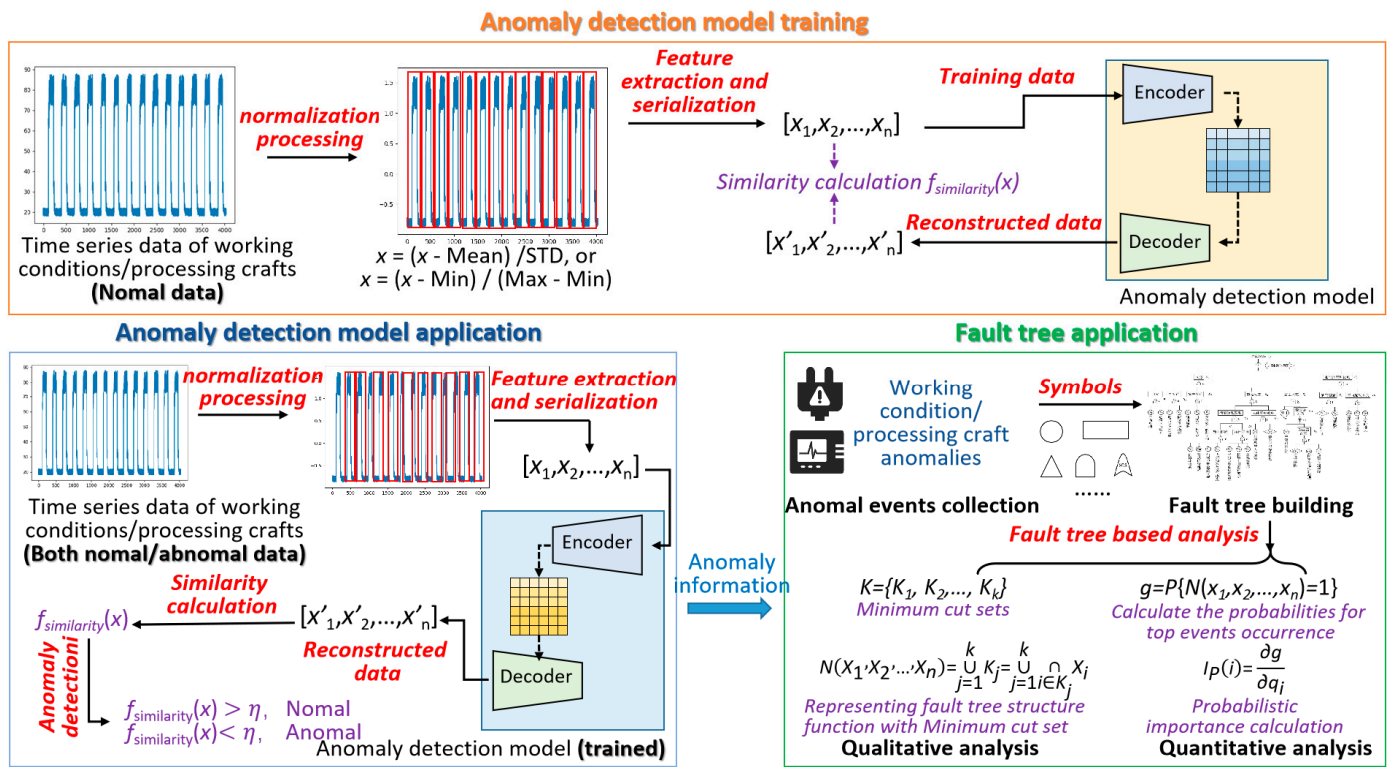


Figure 5. The technique for using deep learning anomaly detection and a fault tree to analyze working condition/processing craft anomalies.

- Convolutional neural network-based polishing quality anomaly analysis: Conventional statistical process control techniques can detect whether there are polishing quality anomalies in the processing line, yet they cannot identify the types of anomalies detected. This technique applies a convolution neural network to identify the types of anomalies by analyzing the patterns of the statistical curves of polishing qualities.
- Bayesian network-based polishing quality anomaly reduction decision-making: After identifying the types of polishing quality anomalies with the previous technique, this technique can be used to identify the detailed causes of the anomalies and thus help carry out anomaly reduction strategies.
- Knowledge graph question-and-answer system-based domain knowledge query: This technique is used to replace commonly used user manuals. The knowledge graph contains the domain knowledge on the operation and maintenance methods of the processing line. Supported by a template-based question-and-answer technique, the system can provide answers to the input queries expressed with natural language. The technique roadmap is illustrated in Figure 6. Firstly, the ontology of the domain knowledge for processing line maintenance was built according to the knowledge from conventional user manuals, experts, and field experience. Secondly, the ontology instantiation-based knowledge graph building method was used to build the domain knowledge graph. Based on the knowledge graph, a template-based question-and-answer system was developed to support natural language-oriented questions and answers. The entire process for building the question-and-answer system includes dictionary building → dictionary-based question sentence embedding → question template set building → decision tree-based question template matching → instantiation of the template with the question sentence → template-based knowledge searching or similarity-based knowledge searching.

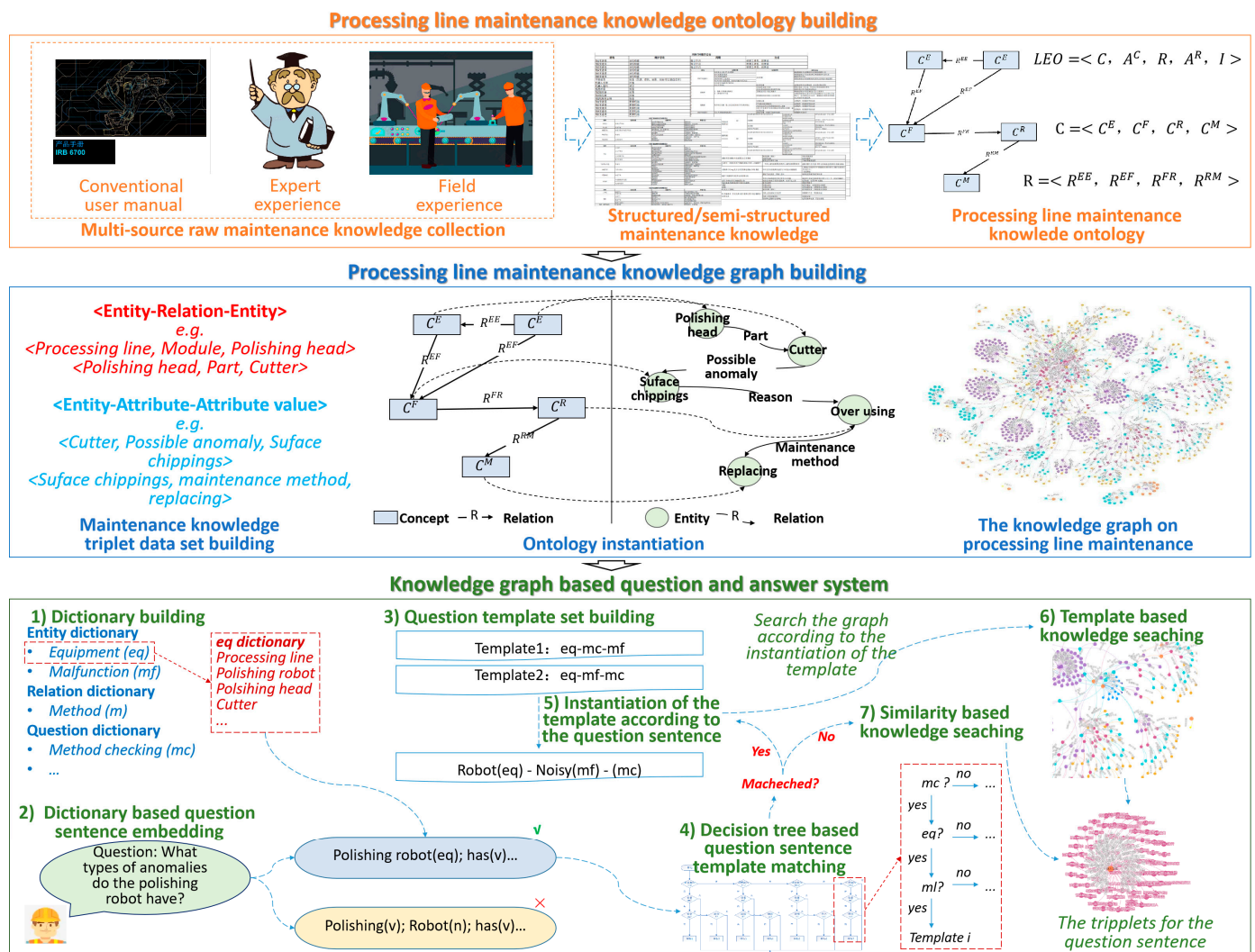


Figure 6. The technique for the knowledge graph-based maintenance knowledge question-and-answer system.

- Ensemble learning-based equipment reliability prediction: The reliability of the processing line is important, because when malfunctions occur the entire production takt is disturbed. This technique is for predicting whether malfunctions would happen in the next production period with a stacking-based ensemble learning framework [27] according to the inputs from routing inspection reports, statistical process control results, working condition/processing craft anomaly detection, etc. The entire process is illustrated in Figure 7. Firstly, the indicators that may influence the reliability of the processing line were collected, and the corresponding dataset was built. For each piece of the datum, the input was the indicator values and the output was the records of malfunctions that happened in the subsequent period of time. Secondly, preprocessing and feature engineering methods were applied to the raw dataset. Based on the preprocessed training data, a stacking-based ensemble learning framework with four base learners was trained. The pseudo-code of the training steps is demonstrated in Table 2.

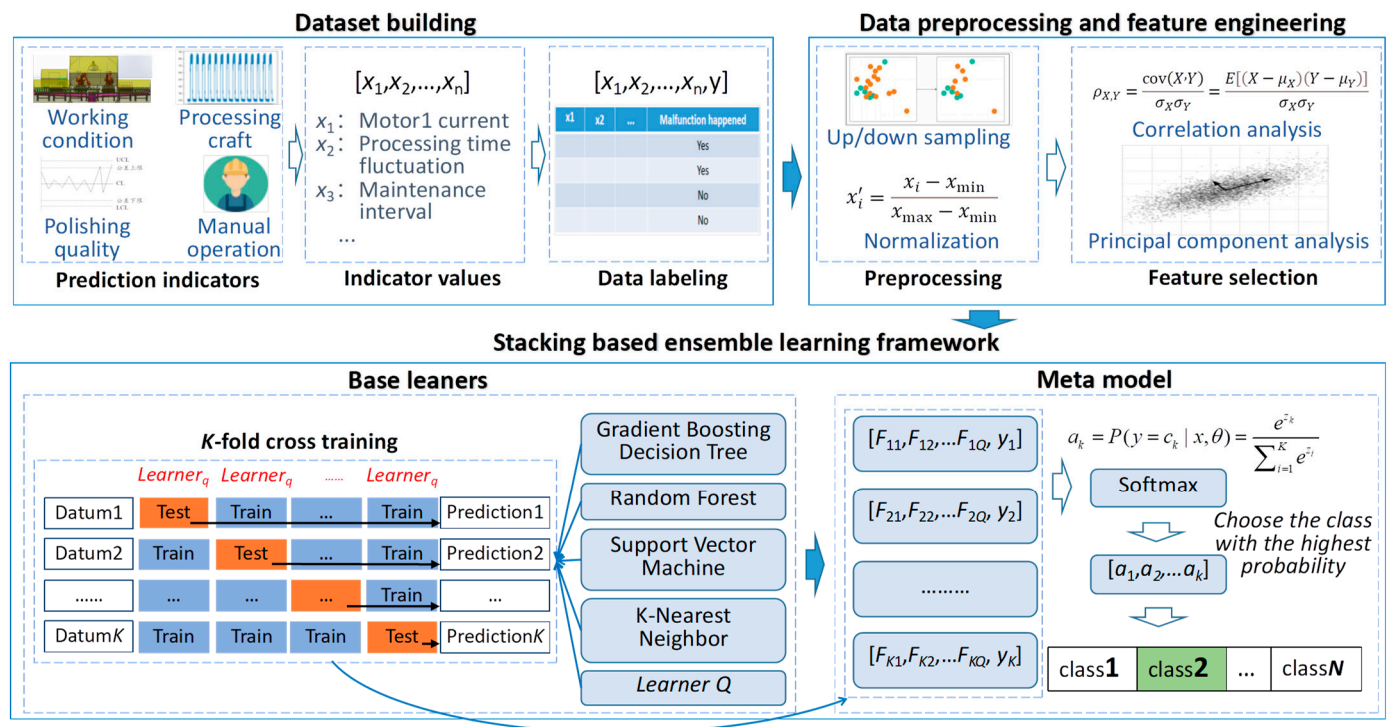


Figure 7. The technique for ensemble learning-based processing line reliability prediction.

Table 2. The training steps of the stacking-based ensemble learning framework.

Algorithm Stacking

Input: training data $D = \{x_i, y_i\}_{i=1}^m$ ($x_i \in \mathbb{R}^n$, $y_i \in Y$) // D is the training set, x_i is the data feature, y_i is the data label, m is the number of training data

Output: an ensemble classifier H

- 1: Step 1: Train the first-level classifiers // the base learners in the first level of the stacking model were trained independently
- 2: **For** $t \leftarrow 1$ to T do // T is the number of base learners
- 3: Train base classifier h_t based on D // h_t is the t th base learner
- 4: **End for**
- 5: Step 2: Construct new data sets from D // construct the training set for the meta model
- 6: **For** $i \leftarrow 1$ to m do
- 7: Construct a new data set that contains $\{x'_i, y_i\}$, where $x'_i = \{h_1(x_i), h_2(x_i), \dots, h_T(x_i)\}$
 // x'_i in the newly built training set is determined by the prediction results from h_1 to h_T
- 8: **End for**
- 9: Step 3: Train the second-level classifier // training the meta-model h'
- 10: Train a new classifier h' based on the newly constructed data set
- 11: **Return** $H(x) = h'(\{h_1(x), h_2(x), \dots, h_T(x)\})$ // the final result is an ensemble learning model consisting of the base learners and the meta model

2.2.4. Visualized Operation and Control Modeling Technique

- Multi-layer/multi-modal event-state dynamic digital twin modeling: This technique was established to support the construction of the dynamic digital twin models for the processing line, and it emphasizes not only the visualization of the working condition/processing craft/polishing quality of the processing line but also its operation/maintenance mechanisms and status.

3. Industrial Application Case

The software of the monitoring and maintenance service system introduced in Section 2 was developed by YX and deployed in YX and QTX (a carbon block production company, which is a customer company in this case) for industrial applications. The software system

was developed based on browser/server architecture, and the users can access the software through browsers. The back end of the system was developed with the *Python* language, *Django* frameworks, and the *PostgreSQL* database. The front-end pages were developed based on the *Vue* framework using *HTML5*, *CSS3*, and *JavaScript*, and the interaction between the front end and the back end is supported by the *Django REST* framework. The software was developed for the *Linux* operating system, and *Docker*-based installation packages were also developed for the deployment of the software in the *Windows* operating system. As mentioned in Section 2.1, the software was developed with a separated configuration/operation architecture; therefore, the configuration and operation systems are two different software systems. However, the core functions of both of the systems were developed in the form of relatively independent WebAPPs. Therefore, the running logic of the two systems can be introduced by showing the working relations among the WebAPPs, as introduced in the following two subsections.

3.1. Configuration System

As mentioned in Section 2.1.1, the configuration system was deployed in the provider company (i.e., YX), and it was designed for generating an encrypted configuration file for the customer company (i.e., QTX) to activate the operation system deployed at QTX. The core workflow of the configuration system can be separated into the three steps below.

3.1.1. Resource Management

This step is for the provider company to manage its template processing lines, sensor networks, and functional WebAPPs in the operation system. The step can be completed by following the five WebAPPs below, as shown in Figure 8.

- General BOM management: This WebAPP is for managing the general design BOM and its corresponding maintenance BOM of the processing line. The specific design BOM and its corresponding maintenance BOM for a specific processing line are subsets of the general design BOM and general maintenance BOM, respectively.
- Template processing line filing: This WebAPP is for establishing the profiles of all the template processing lines.
- Template processing line management: After establishing the profiles of the template processing lines, this WebAPP is for configuring the contents of each template based on the general BOM. By reducing the unrequired components in the general BOM and defining the alternative parts for each remained component, the specific design BOM for a template processing line can be defined. Consequently, the specific maintenance BOM for the template processing line can be determined correspondingly. In addition, the user needs to upload the 3D files of the template processing line for further use in the digital twin WebAPP.
- Alternative sensor management: This WebAPP is for firstly defining all of the optional sensors that can be mounted on the processing lines for different monitoring and maintenance services, and then defining the addresses of the corresponding PLCs and the addresses of the corresponding monitoring datum in the PLCs.
- Operation system WebAPP management: This WebAPP is for defining all of the functional WebAPPs in the operating system together with their URL addresses and other data.

3.1.2. Resource Configuration

This step is for the provider company to define which specific resources are provided to a specific customer in the form of an encrypted configuration file. This step can be completed with the six WebAPPs below one by one, as shown in Figure 9.

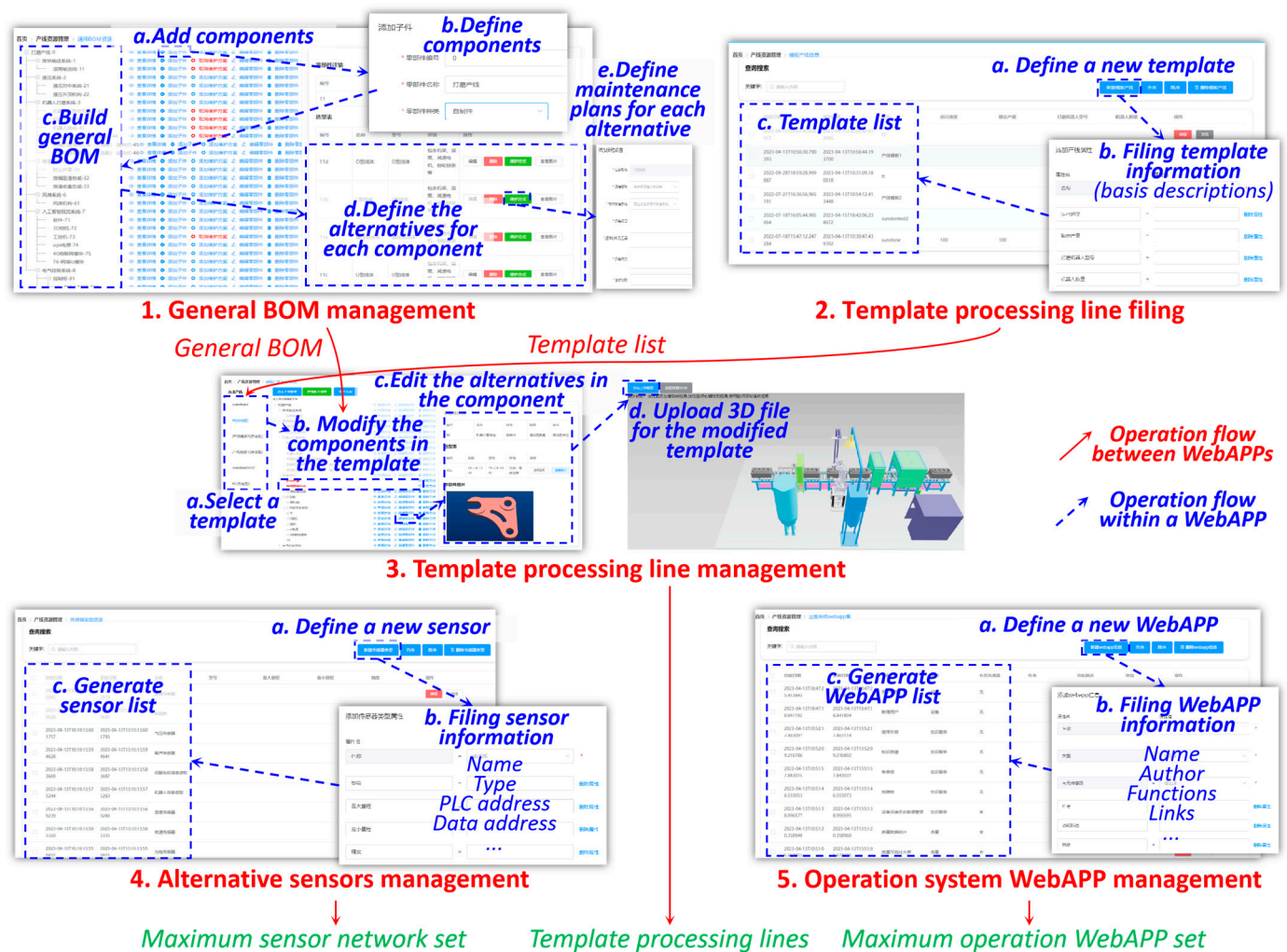


Figure 8. The workflow among the 5 functional WebAPPs for resource management in the configuration system.

- Customer filing: This WebAPP is for establishing the profiles for the customer companies.
- Configuration file initiation: This WebAPP is for creating new/empty configuration files for customers.
- Template-based specific processing line configuration: After selecting a configuration file (for a specific customer), the suitable template processing line for the customer can be selected and then modified according to specific requirements. This way, the specific design BOM of the processing line for the customer can be determined. Correspondingly, the maintenance BOM can be determined.
- Sensor network configuration: After configuring the specific design and maintenance BOMs of the processing line for specific customers, this WebAPP can configure the monitoring and maintenance-related sensors for the processing lines and save the sensor information in the configuration files.
- Operation system WebAPP configuration: After configuring the design/maintenance BOMs and sensors of the processing lines for the customers, this WebAPP can configure the monitoring and maintenance-related functional WebAPPs in the operation system for each customer. It is worth mentioning that there are matching constraints among the design/maintenance BOMs, sensors, and WebAPPs. The constraints were written in this and the previous two WebAPPs.
- Monitoring screen configuration: This WebAPP is for configuring the layout of the contents displayed in the overall working condition monitoring, overall processing craft monitoring, and overall polishing quality monitoring WebAPPs. The three WebAPPs

(introduced in Section 3.2) are displayed on a group of TV screens mounted in the control room of the processing line. To improve convenience during real applications, the configuration is simplified to directly select from predefined templates, each of which defines the contents to be displayed and the layout of the contents.

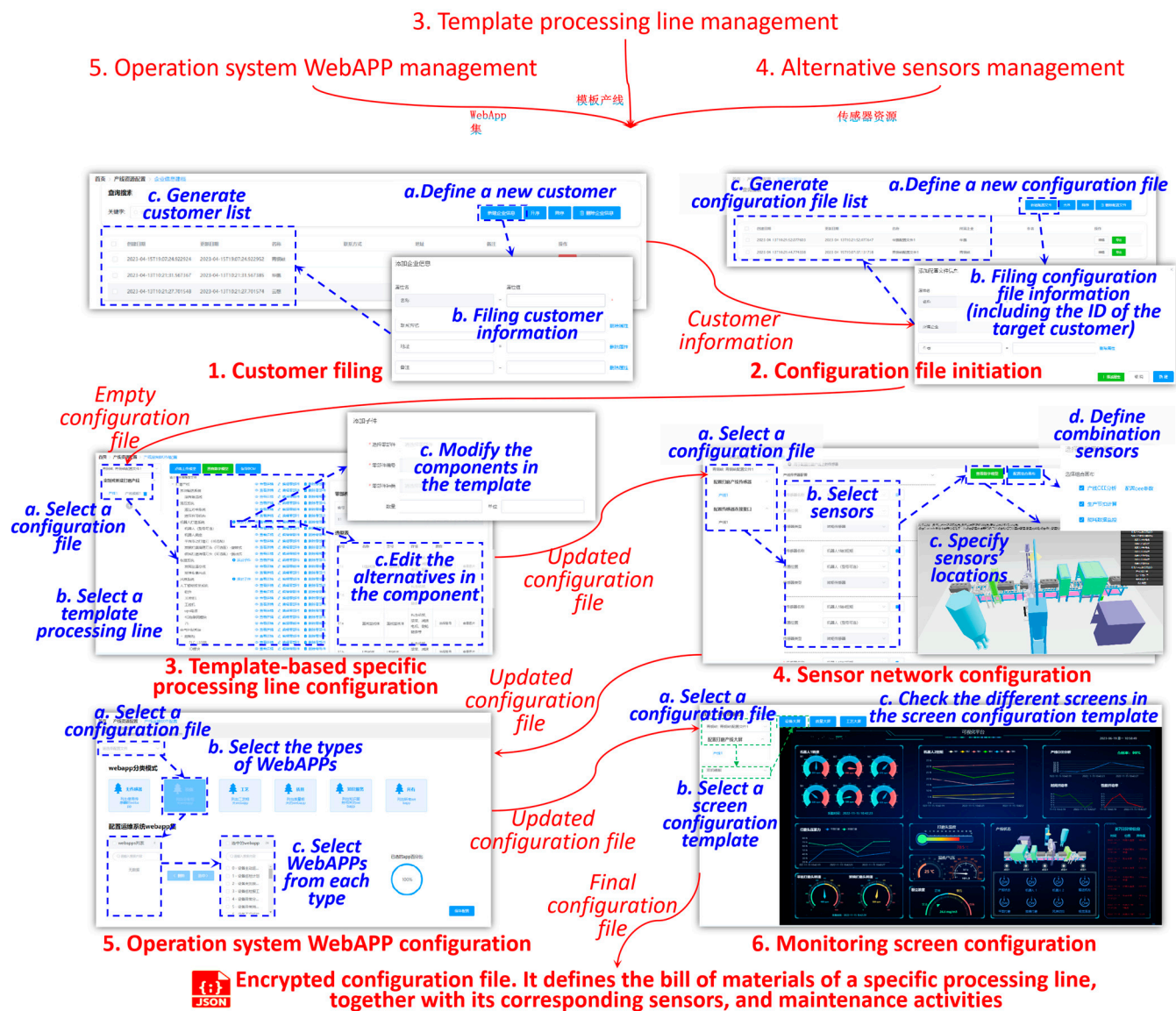


Figure 9. The workflow of the 6 functional WebAPPs for resource configuration in the configuration system.

3.1.3. Knowledge Management

This step is to prepare and manage the domain knowledge in the knowledge service-related WebAPPs in the operating system. Currently, only one WebAPP has been developed for this step, but the interfaces for adding new knowledge management WebAPPs remain.

- **Processing line fault tree management:** This WebAPP is for building the fault trees for the typical anomalies and malfunctions that may occur during the operation of the processing line.

3.2. Operating System

The operating system was deployed in the customer company, and it provides the interface for the customer to acquire the monitoring and maintenance service for the

processing line. The functional WebAPPs in the operation system have to be activated by the encrypted configuration file, and the workflow of the WebAPPs can be separated into four subsections. It is worth mentioning that the WebAPPs in Sections 3.2.1–3.2.3 may operate in parallel. During the process, the WebAPPs in Section 3.2.4 can be invoked to provide knowledge service. The details of the WebAPPs are introduced below. It is also worth mentioning that there should be different roles in the customer company to use these WebAPPs, and a user only has access to the WebAPPs of his/her role. The minimum required roles should be a manager who is responsible for making all kinds of plans and analysis, plus a field engineer who is responsible for directly conducting routine inspection, maintenance, and repair jobs.

3.2.1. Working Condition Monitoring and Maintenance

The WebAPPs in this subsection are for the monitoring and maintenance of the working conditions of the polishing system. The working logic among them is listed below and shown in Figure 10.

- Routine inspection planning: This is for the manager to make the plans for routine inspection and then send the routine inspection assignments to specific field engineers.
- Routine inspection reporting: After the manager has sent the routine inspection assignments to specific field engineers, the field engineers receive the information about the assignments in this WebAPP. By scanning the QR code on the target equipment, the detailed inspection tasks can be read. After the routine inspection tasks have been completed, the field engineer can send reports to the manager through this WebAPP.
- Sudden failure reporting: During the operation of the processing line, any user with any role can send a report if they suddenly find a malfunction or anomaly situation.
- Particular working condition monitoring: Users can check the detailed working condition of each monitoring target through this WebAPP, and the deep learning-based anomaly detection technique introduced in Section 2.2.3 is contained in this WebAPP to intelligently detect whether any anomaly occurs. Reports on the detected anomalies are automatically sent to the next WebAPP.
- Working condition anomaly analysis and dispatch: This WebAPP is for firstly collecting and analyzing the reports on working condition anomalies and malfunctions from the previous three WebAPPs and then sending maintenance task assignments to specific field engineers to handle the anomalies and malfunctions.
- Proactive maintenance planning: When the operation system is deployed in the customer's server for the first time, this WebAPP automatically generates an initial-version life cycle maintenance plan of the processing line according to the maintenance BOMs coded in the configuration file. The manager can check the maintenance plan and send detailed maintenance task assignments to specific field engineers through this WebAPP.
- Working condition maintenance and repairing reporting: Field engineers can check the detailed assignments sent to them from the previous two WebAPPs and then execute corresponding maintenance activities or take care of the anomalies. Afterward, the field engineers can send execution reports to the previous two WebAPPs as feedback to the managers who gave the assignments.
- Historical maintenance records: This WebAPP is for the manager to check the maintenance history of the processing line, including the maintenance activities, the names of the field engineers who executed the maintenance activities, etc.
- Overall working condition monitoring: Different from the WebAPP for particular working condition monitoring, this one provides the overall monitoring information of the processing line in the form of different statistics charts (e.g., line chart of the motor current, pie chart of the polishing quality).
- Digital twin visualization: Driven by the multi-layer/multi-modal event-state dynamic digital twin modeling technique mentioned in Section 2.2.4, this WebAPP is used to provide the visualized digital twin model of the processing line.

- Polishing head lifespan data collection: This WebAPP is for collecting the lifespan working condition data of the cutter heads in polishing heads for future data-driven polishing head malfunction prediction applications.

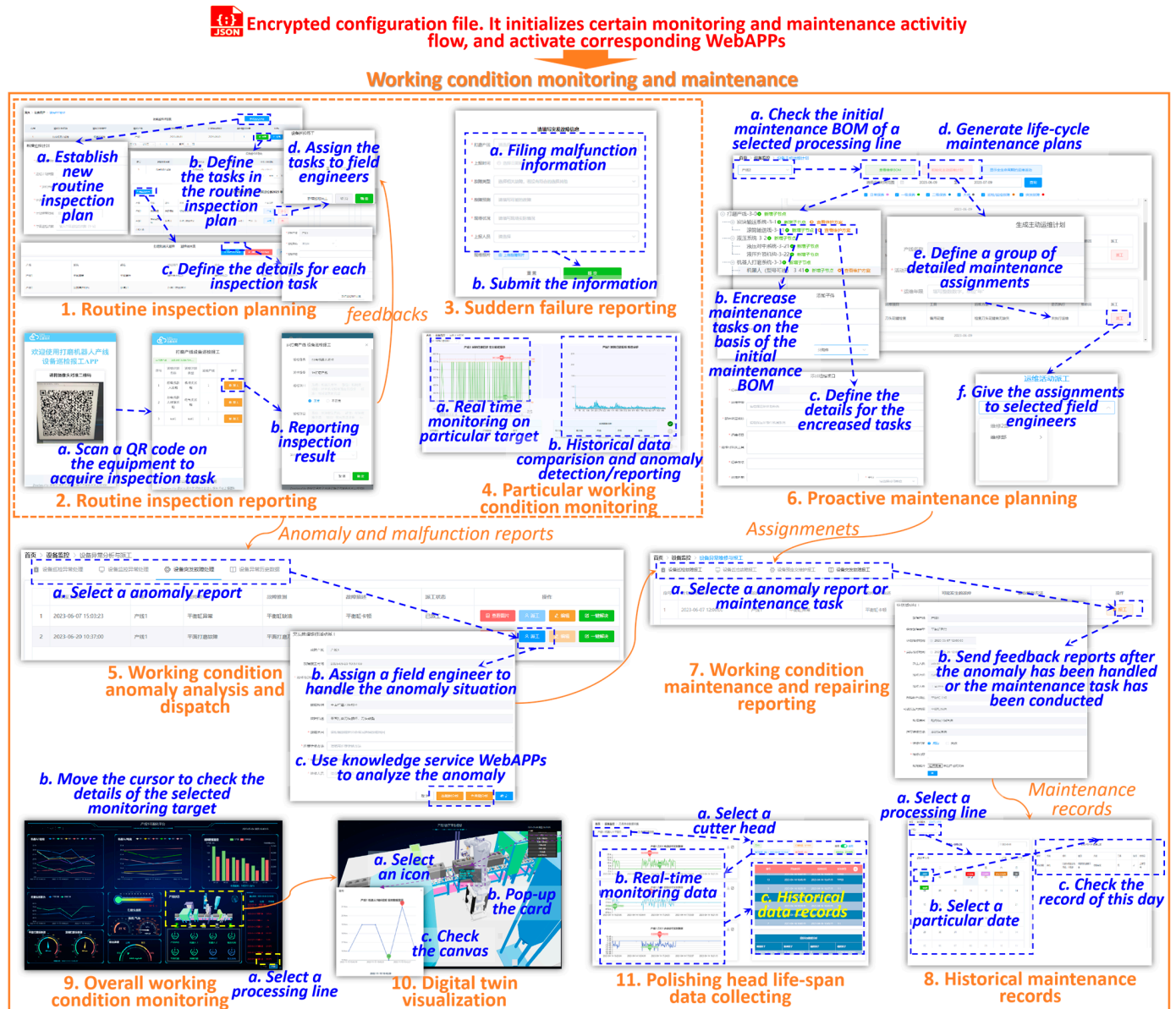


Figure 10. The workflow among 11 functional WebAPPs for working condition monitoring and maintenance in the operating system.

3.2.2. Processing Craft Monitoring and Maintenance

Four WebAPPs were developed for processing craft monitoring and maintenance, and their interaction relationships are listed below and shown in the upper part of Figure 11.

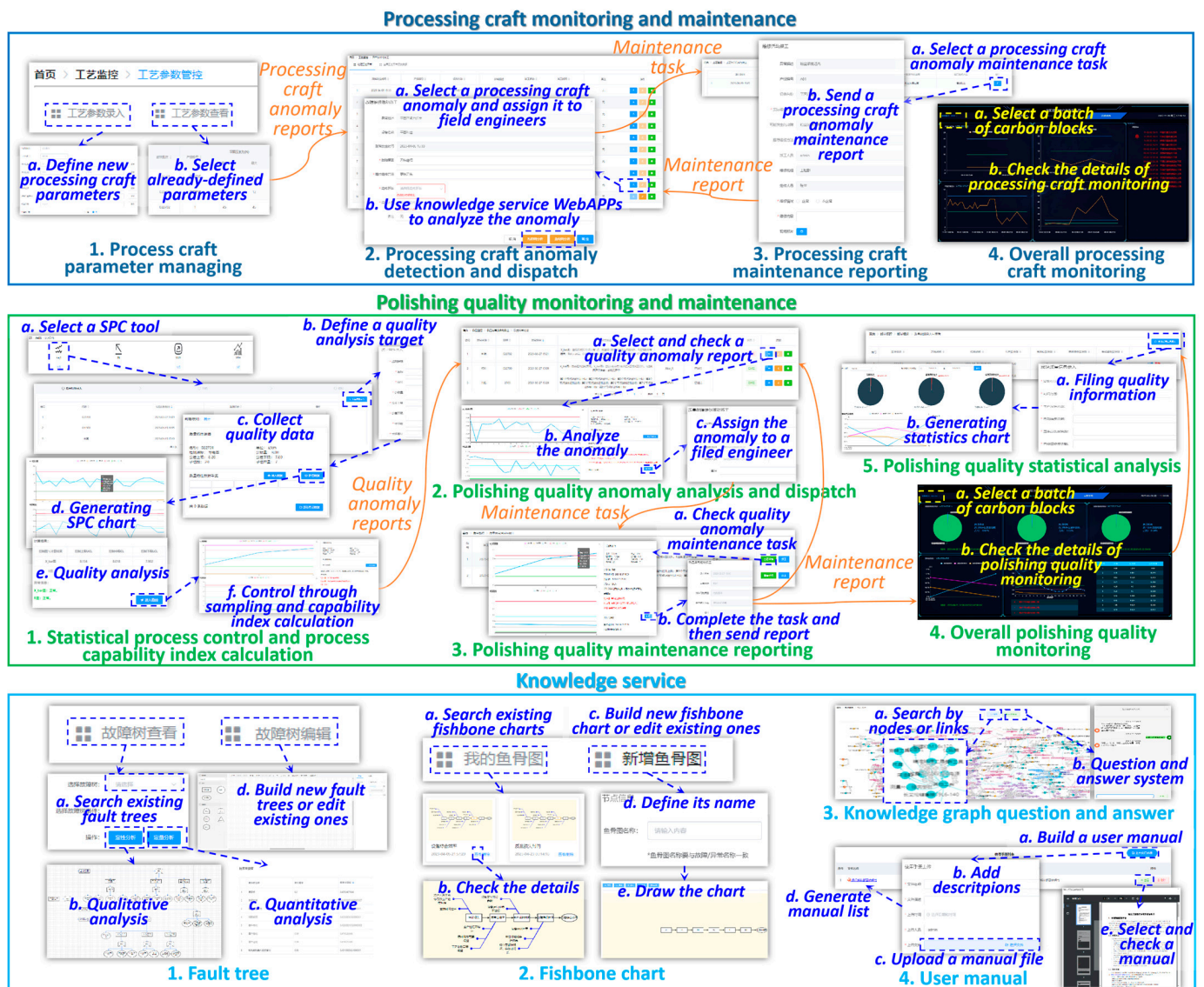


Figure 11. The workflow among functional WebAPPs for processing craft/polishing quality monitoring and maintenance, and knowledge service.

- **Process craft parameter management:** Different batches of carbon blocks usually have different processing crafts, and this WebAPP is for the manager to define the processing craft parameters for different batches of carbon blocks. In addition, if the real-time processing craft parameters are different from the predefined parameters, reports on the anomaly are sent to the next WebAPP.
- **Processing craft anomaly detection and dispatch:** After collecting the reports on processing craft anomalies from the previous WebAPP, this WebAPP is for the manager to analyze the anomaly and assign corresponding maintenance tasks to particular field engineers. During the process, knowledge service WebAPPs can be invoked to support the analysis.
- **Processing craft maintenance reporting:** This WebAPP is for field engineers to receive assignments on processing craft maintenance and send maintenance reports afterward.
- **Overall processing craft monitoring:** This WebAPP is for checking the overall processing craft monitoring status.

3.2.3. Polishing Quality Monitoring and Maintenance

Five WebAPPs were developed for polishing quality monitoring and maintenance, and their interaction relations are listed below and shown in the middle part of Figure 11.

- Statistical process control and process capability index calculation: This WebAPP is for monitoring the processing line by evaluating the polishing quality of the carbon blocks. Firstly, the user can select a tool for the evaluation task (e.g., X Bar R chart [28]), then define a quality analysis target (e.g., the electric conductivity of each carbon block) and collect its monitoring data, and then generate the control chart and analyze whether there are quality anomalies and calculate the process capability index of the processing line. If there are quality anomalies, the anomalies are reported to the next WebAPP.
- Polishing quality anomaly analysis and dispatch: This WebAPP is for the managers to collect all the quality anomaly reports and assign the corresponding maintenance tasks to particular field engineers after preliminary analysis.
- Polishing quality maintenance reporting: Field engineers can check the quality anomaly maintenance tasks assigned to them through this WebAPP and send maintenance reports as feedback after the maintenance tasks have been finished.
- Overall polishing quality monitoring: This WebAPP collects all the polishing quality monitoring and maintenance information from the previous WebAPPs (including real-time quality data, real-time anomaly status, and real-time maintenance feedback) and displays all the information on one monitoring screen.
- Polishing quality statistical analysis: This WebAPP is for displaying different types of statistical analysis charts for one selected quality signal based on its historical monitoring data (e.g., numbers of unqualified carbon blocks per week/month/season in the form of line charts, percentage of unqualified carbon blocks in this week/month/season in the form of pie charts).

3.2.4. Knowledge Service

Four WebAPPs were developed to provide knowledge services during the monitoring and maintenance of working conditions/processing crafts/polishing quality. These WebAPPs work in parallel, as introduced below and shown in the lower part of Figure 11.

- Fault tree: This WebAPP supports the use of the fault trees of the processing lines established by the provider company for both qualitative and quantitative analysis and building new fault trees if the customer company feels it is necessary.
- Fishbone chart: This WebAPP supports using the fishbone charts on different anomaly situations prebuilt by the provider company and building new fishbone charts that contain the knowledge and experience of the engineers of the customer companies.
- Knowledge graph question and answer: This WebAPP was developed based on a knowledge graph that contains domain knowledge related to the operation and maintenance of the processing line. The knowledge graph can be applied through a node/link search and natural language-based question and answer.
- User manual: This WebAPP is mainly an electronic user manual of the processing line. The user can upload, check, and download user manuals through the WebAPP.

4. Discussion and Conclusions

In this paper, an industrial case study on the monitoring and maintenance service system for a robot-driven carbon block polishing service system was established. Firstly, the background and engineering problems that the established monitoring and maintenance service system was trying to solve were introduced. On this basis, the software/hardware architecture of the monitoring and maintenance service system, the strategies for deploying the system in the provider and customer companies, and the working logic among the different sides and levels of the system were introduced. The monitoring and maintenance-related techniques contained in the system together with the relations among the techniques

were introduced. Finally, the configuration and operation workflows of the monitoring and maintenance service system were illustrated.

There are mainly three characteristics of the established monitoring and maintenance service system. Firstly, the system emphasizes comprehensive monitoring and maintenance of the working conditions, processing craft, and polishing quality of the target IPSS (i.e., the robot-based carbon block polishing system), rather than focusing on one or two of the three, like most existing research [18,29]. Secondly, the configuration–operation-separated WebAPP architecture is particularly suitable for the implementation of configurable monitoring and maintenance services. Thirdly, the operating system is supported by multiple functional WebAPPs, each of which contains different maintenance techniques. This way, both conventional and advanced maintenance techniques can be applied in the system, and the system can be conveniently expended by adding new functional WebAPPs containing new maintenance techniques.

The case study can provide real industrial examples for the application of remote monitoring and maintenance service systems on IPSS under the contexts of advanced techniques in Industry 4.0, including the Industrial Internet of Things, cyber–physical systems, digital twins, deep learning, and edge/cloud/fog computing, etc., and this is the main contribution of this paper.

However, the monitoring and maintenance service system in this case study also has limitations. For example, the production takt of the processing line has not been monitored, yet doing so is important for the polishing quality. In addition, some of the functional WebAPPs in the system could be overly complicated for the users; therefore, integration and simplification of the WebAPPs are within our future directions. Furthermore, some of the advanced maintenance techniques are based on deep learning models, and they require large numbers of training data to improve performance. Therefore, the models will be retrained after more data have been collected during the application of the processing line.

Author Contributions: Conceptualization, P.J. and Y.Y.; methodology, P.J., Y.Y. and M.Y.; software, S.S., Y.C., W.Y. and M.Y.; validation, Y.Y. and K.C.; formal analysis, Y.Y. and M.Y.; resources, P.J. and Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, M.Y.; visualization, S.S., Y.C. and W.Y.; supervision, P.J.; project administration, P.J.; funding acquisition, P.J. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Shandong Yantai Development Zone Innovation and Entrepreneurship Leading Team Project (No. 2020CXTD4) and the University-Enterprise Cooperation Project between Xi'an Jiaotong University and Shandong Cloud Imagination Technology Co., Ltd. (No. 20211026).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No additional data are available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, Y.; Yang, M.; Jiang, P. The design of an integrated monitoring and maintenance framework for newly developed equipment: Using industrial robot as example. *IFAC-PapersOnLine* **2022**, *55*, 42–47. [[CrossRef](#)]
2. Yang, M.; Yang, Y.; Jiang, P. A Design Method for Edge–cloud Collaborative Product Service System: A Dynamic Event-state Knowledge Graph-based Approach with Real Case Study. *Int. J. Prod. Res.* **2023**; *in press*. [[CrossRef](#)]
3. Malik, P.K.; Sharma, R.; Singh, R.; Gehlot, A.; Satapathy, S.C.; Alnumay, W.S.; Pelusi, D.; Ghosh, U.; Nayak, J. Industrial internet of things and its applications in industry 4.0: State of the art. *Comput. Commun.* **2021**, *166*, 125–139. [[CrossRef](#)]
4. Kethineni, K.; Gera, P. IoT-based privacy-preserving anomaly detection model for smart agriculture. *Systems* **2023**, *11*, 304. [[CrossRef](#)]
5. Bartocci, E.; Deshmukh, J.; Donzé, A.; Fainekos, G.; Maler, O.; Ničković, D.; Sankaranarayanan, S. Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. In *Lecture Notes in Computer Science*; Bartocci, E., Falcone, Y., Eds.; Springer: Cham, Switzerland, 2018; Volume 10457, pp. 135–175.
6. Tao, F.; Xiao, B.; Qi, Q.; Cheng, J.; Ji, P. Digital twin modeling. *J. Manuf. Syst.* **2022**, *64*, 372–389. [[CrossRef](#)]

7. Serradilla, O.; Zugasti, E.; Rodriguez, J.; Zurutuza, U. Deep learning models for predictive maintenance: A survey, comparison, challenges and prospects. *Appl. Intell.* **2022**, *52*, 10934–10964. [\[CrossRef\]](#)
8. Khan, W.Z.; Ahmed, E.; Hakak, S.; Yaqoob, I.; Ahmed, A. Edge computing: A survey. *Futur. Gener. Comput. Syst.* **2019**, *97*, 219–235. [\[CrossRef\]](#)
9. Zhang, H.; Ma, L.; Sun, J.; Lin, H.; Thürrer, M. Digital twin in services and industrial product service systems: Review and analysis. *Procedia CIRP* **2019**, *83*, 57–60. [\[CrossRef\]](#)
10. Wang, K. Intelligent predictive maintenance (IPdM) system-Industry 4.0 scenario. *WIT Trans. Eng. Sci.* **2016**, *113*, 259–268.
11. Chen, W. Intelligent manufacturing production line data monitoring system for industrial internet of things. *Comput. Commun.* **2020**, *151*, 31–41. [\[CrossRef\]](#)
12. Li, Q.; Yang, Y.; Jiang, P. Remote monitoring and maintenance for equipment and production lines on industrial internet: A literature review. *Machines* **2023**, *11*, 12. [\[CrossRef\]](#)
13. Bin Mofidul, R.; Sabbir, S.H.; Podder, A.K.; Rahman, M.S. Design and implementation of remote controlling and monitoring system for automatic PLC based packaging industry. In Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology, Dhaka, Bangladesh, 3–5 May 2019. [\[CrossRef\]](#)
14. Mourtzis, D.; Vlachou, E.; Milas, N.; Xanthopoulos, N. A cloud-based approach for maintenance of machine tools and equipment based on shop-floor monitoring. In Proceedings of the 48th CIRP Conference on Manufacturing Systems, Ischia, Italy, 24–26 June 2015; Volume 41, pp. 655–660. [\[CrossRef\]](#)
15. Cheng, C.; Zhang, B.; Gao, D. A predictive maintenance solution for bearing production line based on edge-cloud cooperation. In Proceedings of the Chinese Automation Congress, Hangzhou, China, 22–24 November 2019. [\[CrossRef\]](#)
16. Ayvaz, S.; Alpay, K. Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert. Syst. Appl.* **2021**, *173*, 114598. [\[CrossRef\]](#)
17. Cachada, A.; Moreira, P.M.; Romero, L.; Barbosa, J.; Leitno, P.; Geraldcs, C.A.S.; Deusdado, L.; Costa, J.; Teixeira, C.; Teixeira, J.; et al. Maintenance 4.0: Intelligent and predictive maintenance system architecture. In Proceedings of the 23rd IEEE International Conference on Emerging Technologies and Factory Automation, Turin, Italy, 4–7 September 2018. [\[CrossRef\]](#)
18. Zhang, G.; Chen, C.-H.; Cao, X.; Zhong, R.Y.; Duan, X.; Li, P. Industrial Internet of Things-enabled monitoring and maintenance mechanism for fully mechanized mining equipment. *Adv. Eng. Inform.* **2022**, *54*, 101782. [\[CrossRef\]](#)
19. Qi, Q.; Tao, F. A smart manufacturing service system based on edge computing, fog computing, and cloud computing. *IEEE Access* **2019**, *7*, 86769–86777. [\[CrossRef\]](#)
20. Zheng, P.; Wang, Z.; Chen, C.-H.; Khoo, L.P. A survey of smart product-service systems: Key aspects, challenges and future perspectives. *Adv. Eng. Inform.* **2019**, *42*, 100973. [\[CrossRef\]](#)
21. Shihundla, T.B.; Mpofo, K.; Adenuga, O.T. Integrating product-service systems into the manufacturing industry: Industry 4.0 perspectives. *Procedia CIRP* **2019**, *83*, 8–13. [\[CrossRef\]](#)
22. Folgado, F.J.; González, I.; Calderón, A.J. Data acquisition and monitoring system framed in Industrial Internet of Things for PEM hydrogen generators. *Internet Things* **2023**, *22*, 100795. [\[CrossRef\]](#)
23. Tan, S.F.; Samsudin, A. Recent technologies, security countermeasure and ongoing challenges of Industrial Internet of Things (IIoT): A survey. *Sensors* **2021**, *21*, 6647. [\[CrossRef\]](#)
24. Ungurean, I.; Gaitan, N.C. A software architecture for the Industrial Internet of Things—A conceptual model. *Sensors* **2020**, *20*, 5603. [\[CrossRef\]](#)
25. Chen, H. Research and Application of Key Technologies for the Life Cycle Operation and Maintenance of Kneader. Master's Degree, Xi'an Jiaotong University, Xi'an, China, 2022.
26. Jing, T.; Zheng, P.; Xia, L.; Liu, T. Transformer-based hierarchical latent space VAE for interpretable remaining useful life prediction. *Adv. Eng. Inform.* **2022**, *54*, 101781. [\[CrossRef\]](#)
27. Xu, J. Research on Quality Inspection and Production Control of Kneading Process for Anode Paste Used in Carbon Anode Production. Master's Degree, Xi'an Jiaotong University, Xi'an, China, 2023.
28. Hensing, T. X Bar R Control Charts. Available online: <https://sixsigmastudyguide.com/x-bar-r-control-charts/> (accessed on 21 June 2023).
29. Li, D.; Ren, M.; Meng, G. Application of internet of things technology on predictive maintenance system of coal equipment. *Procedia Eng.* **2017**, *174*, 885–889. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.