IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 April 2023, accepted 20 April 2023, date of publication 24 April 2023, date of current version 27 April 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3269802

RESEARCH ARTICLE

A Customized Supervisory Control Approach for Flexible Manufacturing Systems

TING JIAO, (Member, IEEE), AND HONGBING SHI
School of Automation and Software Engineering, Shanxi University, Taiyuan, Shanxi 030006, China
Corresponding author: Ting Jiao (tjiao@sxu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62003199, and in part by the China Scholarship Council under Grant 202108140057.

ABSTRACT To meet the customized requirements of customers is the core orientation to achieve the transition from production-centered manufacturing to service-centered manufacturing. As an important part of smart manufacturing systems in the production and operation level dominated by material flow, flexible manufacturing systems (FMS) have to make several transitions such as incorporating several manufacturing procedures, containing several manufacturing products, and meeting several manufacturing requirements. As the existing research on FMS mainly focuses on fixed manufacturing techniques and production procedures, it is imperative to propose an effective modeling and control approach. In this paper, inspired by the idea of colored Petri nets, based on the manufacturing information of workpieces in the system, we propose a customized supervisory control approach for FMS. Finally, the validity of our approach is illustrated by an FMS satisfying customized requirements. The proposed approach makes control decisions by dynamically updating the token information of the current state of the supervisor, and also by combining the information of the binding queue and the customized information of workpieces, which does not increase the scale of the supervisor.

INDEX TERMS Discrete-event systems, supervisory control theory, flexible manufacturing systems, colored Petri nets, customization.

I. INTRODUCTION

Flexible manufacturing systems (FMS) have the characteristics of flexibility, dexterity, and collaboration, and are an important part of the production and operation level of intelligent manufacturing systems dominated by material flow [1]. Modern information technology has brought profound changes to the production, management and marketing modes of FMS. Moreover, the production concept of "customer driving demand, and service creating value" has become increasingly prominent. The customized needs of customers exhibit many forms, such as:

- In the semiconductor manufacturing industry, which is well-known for its precision, it is necessary to determine the number of polishing times for wafers according to the different processing accuracy requirements of customers [2].
- When processing high-cost workpieces, the production process of the workpiece needs to be determined according to the number of reprocessing anticipated by each customer [3].
- In the process of personalized processing based on customer's requirements, it is necessary to select the corresponding product filling formula and product combination method according to the specified needs of customers, and constantly adjust the processing process dynamically [4].

To meet the above customized needs, FMS need to make the following upgrades:

- FMS should be compatible with the coexistence of multiple processing procedures. The production of a certain product often requires multiple components to work together [5]. In FMS, the number of these components is relatively fixed, but the customization of processing procedures will cause multiple processing procedures

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwu Li.

to coexist in the system. It is necessary to study how to ensure the correct operation of multiple processing procedures simultaneously. Besides, it should be able to respond to the dynamic changes of customer's needs.
- FMS should have the ability to accommodate a variety of products simultaneously. Due to the customized requirements of customers, the types of products produced by FMS will be more abundant. For example, customers have different requirements on the processing accuracy of products, inducing that the numbers of processing times of different products are also different [6]. Therefore, there exist some differences in the processing flow of the same type of products, and it is necessary to study how to accommodate the processing of multiple products simultaneously.
- FMS should meet a variety of processing needs. In the past, the processing requirements mainly considered the causal logic involving the sequence [7]. However, in order to make the operation of the FMS more reasonable and more efficient, the temporal logic should also be considered, such as how to ensure that multiple products have equal processing opportunities [8], [9], [10], [11].

For the modeling and control method of FMS, the existing research work mainly includes the following two aspects:

(1) Real-time scheduling problem of FMS

Qiao et al. employed Petri nets as the modeling tools, and took robotic cluster tools as the main research object. They studied the scheduling problem of wafer manufacturing, and analyzed the main performance parameters of wafer dwell delay and robot waiting time [12], [13]. The revisiting problem in the wafer manufacturing process has also been widely studied. Wu et al. proposed a series of analysis methods for the revisiting problem based on Petri nets constrained by the limited residence time and fluctuating activity time [14], [15]. Jia et al. conducted a real-time analysis on the assembly system fulfilling small batch customized production tasks [16]. Kimble et al. proposed a benchmark protocol for evaluating small parts robotic assembly systems [17]. Li et al. proposed a scheduling scheme to realize customized production based on the bat algorithm [18]. Wang et al. studied the FMS scheduling problem subject to no-wait constraints via Petri nets and heuristic search [19].

(2) Supervisory control problem of FMS

A comprehensive literature review for the deadlock control of automated manufacturing systems based on Petri nets refers to reference [20]. Additionally, Chen et al. studied the optimal deadlock control strategy of FMS and its application problem based on Petri nets [21], [22], [23]. Yue et al. studied the resource fault and buffer space allocation control problem of automatic manufacturing systems [24]. Liu et al. proposed the design method of optimal Petri net controller for FMS containing key resources [25]. Huang et al. proposed crucial marking/transition-separation instances (MTSIs) allowing designers to employ much fewer MTSIs to deal with deadlocks for FMS [26]. Li et al. proposed a deadlock prevention approach based on structure reuse of Petri nets for FMS [27]. Hu et al. proposed a deadlock-free control policy of FMS with flexible routes and assembly operations using Petri nets [28]. Liu et al. proposed a live Petri net controller synthesis approach based on a controllable siphon basis to achieve deadlock prevention for FMS [29]. Luo et al. proposed a Petri nets-based deadlock avoidance policy for FMS with assembly operations and multiple resource acquisition [30]. Liu et al. proposed a deadlock controller synthesis approach for FMS based on the max-controllability of siphons [31]. Duan et al. proposed a deadlock prevention policy for FMS modeled by Petri nets using structural analysis [32]. Du et al. proposed a control policy for the robust deadlock avoidance and control of FMS [33]. Luo et al. proposed a robust deadlock avoidance policy for FMS with multiple unreliable resources [34]. Bashir et al. proposed an optimal supervisory control approach for FMS with zero restrictions of system operations [35]. Lu et al. proposed an efficient method of deadlock detection and recovery for FMS by resource flow graphs [36]. Fan et al. proposed event circuit structures for deadlock avoidance in FMS [37]. Overall, the FMS targeted by these studies mainly produce the same type of products, so they are less concerned with how to meet the customized needs.

As flexible manufacturing evolves towards customer-driven demands, how to meet customized demands has gradually attracted extensive attention in the academic community.

In order to meet the customized requirements of distinguishing the number of reprocessing of each workpiece, Cury et al. employed finite state automata (FSA) as the modeling tool, and proposed the concept of distinguisher to identify the number of substandard products [3]. The distinguisher identifies some vital events (such as the events corresponding to buffer overflow and underflow) individually, and then directly establishes the FSA model of the specification based on these identified events. Although this method can greatly simplify the modeling of the specification, the complexity of modeling is transferred to the modeling of the plant. Therefore, the overall complexity of the resulting control scheme is not reduced. In order to reduce the complexity of the modeling, Cury et al. proposed a model approximation method, which reduces the number of vital events being identified and makes the information contained in the plant to be controlled more general. Although this method can reduce the state numbers of the plant and the controller to a certain extent, the solution obtained by this method may not be optimal (i.e., supremal in the context of the supervisory control theory) [38], [39], [40], [41]. Teixeira et al. applied the distinguisher to the local modular control of discrete-event systems (DES), which uses the idea of decentralization and modularization to make the structure of the controller more concise and more transparent [42]. Mahdavinezhad et al. proposed a method to use causal output maps to extract part of the information in the supervisor to distinguish events with the same name. The results are applied to those hybrid systems which can be approximated as DES [43]. Ushio et al. studied the supervisory control problem of DES using Mealy automata with uncertain output function [44]. Yin proposed a model transformation method based on the output function of Mealy automata [45]. However, these methods are still mainly aimed at the processing of the same type of products, and the modeling process is still complicated.

It can be seen from the above analyses that the key to realizing the customized supervision and control of the FMS lies in the need to dynamically record and update the workpiece information in the system, in which the customized requirements are contained. In colored Petri nets, the update of information in the system is achieved by employing the variable expressions in high-level programming languages [46]. Inspired by the method of updating the token information of the controller's current state in the colored Petri nets, this paper introduces the token set in each state of the supervisor to record the workpiece being processed in this state. Then, the supervisor makes control decisions according to both the supervisory control strategy and the token set, thereby realizing the customized supervisory control.

The rest of this paper is organized as follows. Section II introduces the relevant basic knowledge of the supervisory control theory for DES. Section III elaborates the formal description and implementation process of the customized supervisory control. Section IV presents an example to demonstrate the validity of the proposed approach. Section V presents our conclusions.

II. PRELIMINARIES

A. DISCRETE-EVENT SYSTEMS

The formal structure of a DES to be controlled is a deterministic finite state automaton, say

$$G = (Q, \Sigma, \delta, q_0, Q_m).$$

Here, Σ is a collection of symbols representing asynchronous events, Q is the *state set*, $\delta : Q \times \Sigma \rightarrow Q$ is the (partial) *transition function*, q_0 is the *initial state*, and $Q_m \subseteq Q$ is the subset of *marked states*. For brevity, transition $\delta(q, \sigma) = q'$ is often denoted as $q \xrightarrow{\sigma} q'$.

Usually, Σ is divided into two parts: the subset of controllable events Σ_c and the subset of uncontrollable events Σ_u, with $\Sigma = \Sigma_c \dot{\cup} \Sigma_u$, where symbol $\dot{\cup}$ represents a disjoint union of sets.

For any language $L \subseteq \Sigma^*$, the prefix closure consisting of the prefixes of all strings in L is written as

$$\bar{L} = \{t \in \Sigma^* | tu = s, u \in \Sigma^*, s \in L\}.$$

The *closed behavior* of plant G is defined as

$$L(G) = \{s \in \Sigma^* | \delta(q_0, s)!\},$$

where the notation $\delta(q_0, s)!$ means that $\delta(q_0, s)$ is defined.

The *marked behavior* of plant G is defined as

$$L_m(G) = \{s \in \Sigma^* | \delta(q_0, s) \in Q_m\}.$$

The inclusive relation between the closed behavior and the marker behavior of G is $\emptyset \subseteq L_m(G) \subseteq L(G)$.

DES G is *nonblocking* if and only if $\overline{L_m(G)} = L(G)$.

B. SUPERVISORY CONTROL THEORY

Supervisory control theory (SCT) deals with the control of DES [7], [8]. For DES $G = (Q, \Sigma, \delta, q_0, Q_m)$, the supervisor can only disable the controllable events in Σ, and the uncontrollable events are enabled by default. Each subset of events to be enabled (adjoining all the uncontrollable events) is a *control pattern*, and the set of all control patterns is denoted by $\Gamma = \{\gamma \in \mathrm{Pwr}(\Sigma) | \gamma \supseteq \Sigma_u\}$, where $\mathrm{Pwr}(\cdot)$ represents a power set [7].

Use the mapping $V : L(G) \rightarrow \Gamma$ to describe the supervisory control of G, and denote the supervisory control relationship (G, V) between G and V as V/G, meaning that "G is under the supervision of V".

The closed behavior $L(V/G) \subseteq L(G)$ of the language generated by G under the supervision of V satisfies

(1) Empty string $\varepsilon \in L(V/G)$;

(2) If $s \in L(V/G)$, $\sigma \in V(s)$ and $s\sigma \in L(G)$, then $s\sigma \in L(V/G)$;

(3) All other strings do not belong to $L(V/G)$.

The marked behavior of the language generated by G under the supervision of V is $L_m(V/G) = L(V/G) \cap L_m(G)$.

The empty set satisfies $\emptyset \subseteq L_m(V/G) \subseteq L_m(G)$. If $\overline{L_m(V/G)} = L(V/G)$, then V is said to be *nonblocking* with respect to G [7].

Language $K \subseteq \Sigma^*$ is controllable with respect to G if and only if

$$(\forall s, \sigma) s \in \overline{K} \ \&\ \sigma \in \Sigma_u \ \&\ s\sigma \in L(G) \Rightarrow s\sigma \in \overline{K}.$$

For a given specification $E \subseteq \Sigma^*$, the set of all sublanguages of E controllable with respect to G can be expressed as

$$\mathcal{C}(E) = \{K \subseteq E | K \text{ is controllable with respect to } G\}.$$

Since each element in $\mathcal{C}(E)$ is closed under union and empty set $\emptyset$ belongs to $\mathcal{C}(E)$, there exists a supremal element in $\mathcal{C}(E)$, denoted as $K_{\mathrm{sup}} = \sup \mathcal{C}(E)$. K_{sup} can be computed by procedure supcon[1] in TCT[2], and the result is the supervisor [7].

Let supervisor $S := (Q_S, \Sigma, \delta_S, q_{0S}, Q_{mS})$ represent the supremal controllable sublanguage K_{sup}. For a given string $s \in K_{\mathrm{sup}}$, let $q := \delta_S(q_{0S}, s)$ denote the *current state* of supervisor S under the occurrence of string s. Define *the set of events enabled at current state* q by map $\mathcal{E} : Q_S \rightarrow \mathrm{Pwr}(\Sigma)$ with

$$\mathcal{E}(q) := \{\sigma \in \Sigma | \delta_S(q, \sigma)!\}.$$

[1]DES3 = supcon(DES1, DES2) is a trim automaton for the supremal controllable sublanguage of the marked legal language generated by DES2 with respect to the plant DES1 [7].

[2]TCT is a software package for the synthesis of supervisory controls for DES, which can be freely downloaded from https://www.control.utoronto.ca/cgi-bin/dlxptct.cgi

III. CUSTOMIZED SUPERVISORY CONTROL

Assume that the plant and the specification of the DES to be controlled are represented by symbols G and E respectively. When the customization is not considered, the corresponding supervisor can be computed by procedure supcon as follows:

$$S := \text{supcon}(G, E).$$

In order to realize the customized production of workpieces, we assume that each component in the system identifies the identity information of each workpiece W_i through radio frequency identification technology (RFID), denoted as ID(i). Correspondingly, in each state q of supervisor S, token set $T(q)$ records the workpiece identity information in this state.

Divide the events in workpiece W_i into three categories: Σ_i^{in} is the set of events representing workpiece W_i entering the system, Σ_i^{out} is the set of events representing workpiece W_i leaving the system, and Σ_i^{extra} represents other events in workpiece W_i.

Assume that transition $q \xrightarrow{\sigma} q'$ occurs in the supervisor to process workpiece W_i. Then the token sets of states q and q' are updated according to the following rules:

(1) If $\sigma \in \Sigma_i^{\text{extra}}$, then

$$T(q) = T(q) \setminus \{\text{ID}(i)\},$$
$$T(q') = T(q') \cup \{\text{ID}(i)\}.$$

(2) If $\sigma \in \Sigma_i^{\text{in}}$, then

$$T(q) = T(q),$$
$$T(q') = T(q') \cup \{\text{ID}(i)\}.$$

(3) If $\sigma \in \Sigma_i^{\text{out}}$, then

$$T(q) = T(q) \setminus \{\text{ID}(i)\},$$
$$T(q') = T(q').$$

If the closed behavior and the marked behavior of workpiece W_i satisfy conditions

$$L(W_i) \subseteq L(S),$$
$$L_m(W_i) \subseteq L_m(S),$$

we say that the *customized requirements of workpiece W_i match supervisor S*.

Remark 1: If the customized requirements of workpiece W_i do not match supervisor S, to ensure nonblockingness, we have to modify W_i until its customized requirements match supervisor S.

On the premise that the workpieces' customized requirements match supervisor S, the customized supervisory control architecture is shown in Fig. 1, where it is assumed that there are n workpieces to be processed in the current system. Compared with the Ramadge-Wonham's supervisory control architecture, the customized supervisory control architecture proposed in this paper adds automata $W_i, i \in \{1, \cdots, n\}$ describing the customized requirements and queue $\mathcal{Q}$ managing the sequence of events.

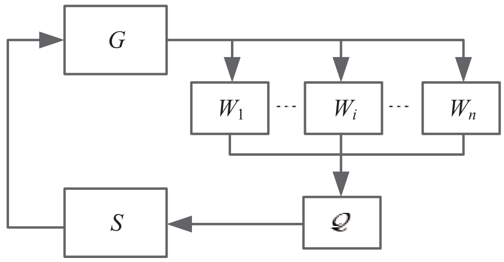

FIGURE 1. Schematic of the architecture of customized supervisory control.

The implementation process of the customized supervisory control is as follows.

When workpiece W_i is going to enable a controllable event σ_c, binding (ID(i), σ_c) is generated. If the binding already exists in queue $\mathcal{Q}$, this binding is ignored; otherwise, binding (ID(i), σ_c) is added to the end of queue $\mathcal{Q}$.

In the current state of the supervisor, the events to be enabled are divided into two cases:

(1) The enabled controllable event σ_c is taken from the head binding (ID(i), σ_c) of queue $\mathcal{Q}$, and requires that the workpiece identity information ID(i) of this bingding should belong to the token set of the current state of the supervisor. After event σ_c occurs, the current state of workpiece W_i, the current states of supervisor S, and the current state of plant G are updated. In addition, the relevant token sets are updated according to the rules described above.

(2) The uncontrollable event σ_u is enabled according to the actual processing flow of the component. The current state of the workpiece, the supervisor and the plant corresponding to uncontrollable event σ_u are updated accordingly. In addition, the relevant token sets are updated according to the rules described above.

After the current state of the supervisor is updated by the occurrence of an event, the events in this new current state are enabled according to the above rules, thereby propelling the continuous operation of the controlled system.

Theorem 1: If the customized requirements of each workpiece W_i match the supervisor S, then the resultant controlled behavior is nonblocking.

Proof: To prove the nonblocking behavior of the controlled system, we only need to prove that the controlled behavior is nonblocking after introducing the closed behavior $L(W_i)$, the marked behavior $L_m(W_i)$, and queue $\mathcal{Q}$.

In the current state q of supervisor S, there are two types of events that can occur. For controllable event σ_c, it needs to satisfy $\sigma_c \in \mathcal{E}(q)$ and σ_c is taken from the head element of queue $\mathcal{Q}$. For uncontrollable event σ_u, $\sigma_u \in \mathcal{E}(q)$ needs to be satisfied.

As $L_m(S)$ is controllable with respect to G and $L_m(S)$ is $L_m(G)$-closed, i.e., $L_m(S) = L(S) \cap L_m(G)$, under the action of the control patterns corresponding to supervisor S, the controlled behavior of the system is nonblocking by Theorem 3.3 in [7]. From the above analysis, it can be

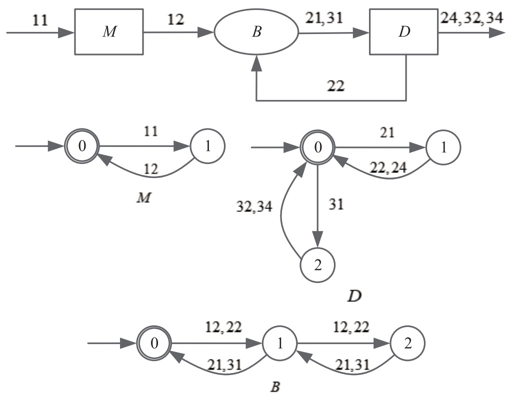

FIGURE 2. Schematic of FMS with workpiece reprocessing.

seen that by introducing workpieces' closed behavior $L(W_i)$, marked behavior $L_m(W_i)$, and queue $\mathcal{Q}$, the set of events enabled at the current state of supervisor S is a subset of a given control pattern. Hence, the resultant controlled behavior is also nonblocking. ■

Theorem 1 ensures that the proposed customized supervisory control architecture can realize the nonblocking operation of the controlled system. The basic idea involved in Theorem 1 is similar to the idea of model abstraction and refinement in model checking. Namely, if the original system meets the requirements, the new system obtained by refining the original system still meets the established requirements [47].

IV. EXAMPLE OF AN FMS WITH WORKPIECE REPROCESSING

This paper takes the FMS with workpiece reprocessing shown in Fig. 2 as an example to demonstrate the specific implementation process of the customized supervision and control. The FMS includes a processing unit M, a buffer B and a detection unit D. The processing unit M takes workpieces from the outside of the system. The workpiece is processed (represented by event 11), and placed in buffer B after processing (represented by event 12). The detection unit D is responsible for reprocessing and detection, and it has two working modes. Mode 1 takes the workpiece from buffer B (represented by event 21). If the reprocessing meets the requirements, the workpiece will be output to the system (represented by event 24). Otherwise, the workpiece will be put back into buffer B (represented by event 22). In Mode 2, the workpiece taken from buffer B (represented by event 31) will not be returned to the system after reprocessing, and the workpiece will be output from the system if it passes the inspection (represented by event 34); otherwise, the workpiece is discarded (represented by event 32). Assume that the buffer capacity is 2, and the specification that needs to be satisfied is to prevent buffer B from overflow and underflow. The detailed state transition diagram of each component in the system is shown in Fig. 2.

The FMS in this example needs to meet the customized requirements about the maximum number of reprocessing of each workpiece specified by the user. Without loss of generality, we assume that the user needs to process 2 workpieces, and the number of reprocessing of each workpiece is required not to exceed 2 times.

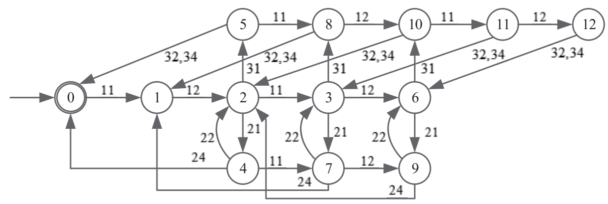

FIGURE 3. State transition diagram of supervisor S.

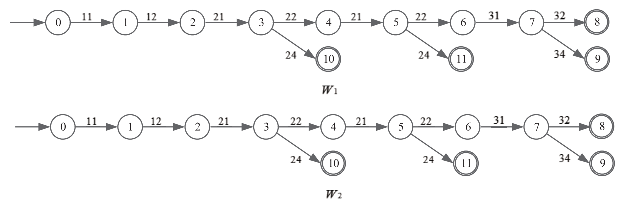

FIGURE 4. State transition diagrams of workpieces W_1 and W_2.

According to the execution process of the customized supervisory control presented in Section III, the supervisor without customization is computed first, and its state transition diagram is shown in Fig. 3.

The behaviors of the workpieces that meet the customized requirements are shown in Fig. 4. Each state transition diagram describes three situations: the workpiece meets the requirements after processing once, meets the requirements after processing twice, and still fails to meet the requirements after processing twice. As $L(W_i) \subseteq L(S)$, $L_m(W_i) \subseteq L_m(S)$, the customized requirements of workpieces W_1 and W_2 match the supervisor.

Assume that the current states of the supervisor S, workpieces W_1 and W_2 are represented by q_S, q_{W_1}, q_{W_2}, respectively. If workpiece W_1 is going to enable event 11, binding $(\mathrm{ID}(W_1), 11)$ will enter queue $\mathcal{Q}$. After event 11 is enabled, the current states of supervisor S, workpieces W_1 and W_2 are updated to $q_S = 1, q_{W_1} = 1, q_{W_2} = 0$, respectively, and the token set corresponding to q_S is $T(1) = \{\mathrm{ID}(W_1)\}$. Then, supervisor S waits for uncontrollable event 12 to occur. After event 12 is enabled, the current states of supervisor S, workpieces W_1 and W_2 are updated to $q_S = 2, q_{W_1} = 2, q_{W_2} = 0$, respectively, and the token set corresponding to q_S is $T(2) = \{\mathrm{ID}(W_1)\}$.

Similarly, assume that workpiece W_2 enters the system and is processed by processing unit M. Then the current states of supervisor S, workpieces W_1 and W_2 are updated to $q_S = 6, q_{W_1} = 2, q_{W_2} = 2$, respectively, and the token set corresponding to q_S is $T(6) = \{\mathrm{ID}(W_1), \mathrm{ID}(W_2)\}$.

If workpiece W_1 is returned to buffer B by detection unit D, then the current states of supervisor S, workpieces W_1 and W_2 are updated to $q_S = 6, q_{W_1} = 4, q_{W_2} = 2$, respectively, and the token set corresponding to q_S is $T(6) = \{\mathrm{ID}(W_1), \mathrm{ID}(W_2)\}$.

Subsequently, if workpiece W_1 is returned to buffer B by detection unit D again, the current states of supervisor S, workpieces W_1 and W_2 are updated to $q_S = 6, q_{W_1} = 6, q_{W_2} = 2$, respectively, and the corresponding token set is $T(6) = \{\mathrm{ID}(W_1), \mathrm{ID}(W_2)\}$.

To this end, workpiece W_1 can no longer enable event 21, because in the current state of workpiece W_1, only binding $(\mathrm{ID}(W_1), 31)$ can be generated to ensure the user's customized requirement that workpiece W_1 can be reprocessed at most 2 times.

The aforementioned customized supervisory control approach is suitable for the situation that more workpieces enter and leave the system, or other customized workpiece behaviors satisfy the customized requirements of workpiece W_i and supervisor S.

Compared with the approaches in the literature, the customized supervisory control approach proposed in this paper does not need to expand the scale of the supervisor. The events in each workpiece are enabled by dynamically updating the token information of the current state of the supervisor, the binding queue information and the customized processing information of the workpiece.

V. CONCLUSION

This paper proposes a customized supervisory control approach for FMS. In this approach, the customized requirements of workpiece processing are modeled by deterministic finite state automata. During the operation of the system, the proposed customized supervisory control approach makes control decisions by retrieving the binding information of the controllable events to be enabled from the queue, and by combining the current state of the supervisor with the controllable event binding information.

In future research, the approach proposed in this paper is to be extended to DES modeled by extended finite state automata [48], state tree structures [49] and relabeled DES [50] to realize corresponding customized supervisory control.

REFERENCES

[1] F. Y. Wang, Y. C. Gao, and X. Q. Shang, "Parallel manufacturing and industries 5.0: From virtual manufacturing to intelligent manufacturing," *Sci. Technol. Rev.*, vol. 36, no. 21, pp. 10–22, 2018.

[2] Y. Qiao, N. Wu, F. Yang, M. Zhou, Q. Zhu, and T. Qu, "Robust scheduling of time-constrained dual-arm cluster tools with wafer revisiting and activity time disturbance," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 6, pp. 1228–1240, Jun. 2019.

[3] J. E. R. Cury, M. H. de Queiroz, G. Bouzon, and M. Teixeira, "Supervisory control of discrete event systems with distinguishers," *Automatica*, vol. 56, pp. 93–104, Jun. 2015.

[4] F. Wang, S. Shu, and F. Lin, "Robust networked control of discrete event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 4, pp. 1528–1540, Oct. 2016.

[5] Y. Shi, Q. Han, W. Shen, and H. Zhang, "Potential applications of 5G communication technologies in collaborative intelligent manufacturing," *IET Collaborative Intell. Manuf.*, vol. 1, no. 4, pp. 109–116, Dec. 2019.

[6] J. Wang, C. Pan, H. Hu, L. Li, and Y. Zhou, "A cyclic scheduling approach to single-arm cluster tools with multiple wafer types and residency time constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1373–1386, Jul. 2019.

[7] W. M. Wonham and K. Cai, *Supervisory Control of Discrete-Event Systems*. New York, NY, USA: Springer, 2019.

[8] D. Z. Zheng and Q. C. Zhao, *Discrete-Event Dynamic Systems*. Beijing, China: Tsinghua Univ. Press, 2001.

[9] T. Jiao and Z. G. Liu, "Study on the supervisory control of discrete-event systems incorporating components with several working modes," *Control Theory Appl.*, vol. 37, no. 3, pp. 534–539, 2020.

[10] R. Zhang and K. Cai, "Supervisor localization of discrete-event systems with infinite behavior," *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 361–366, 2018.

[11] A.-K. Schmuck, T. Moor, and R. Majumdar, "On the relation between reactive synthesis and supervisory control of non-terminating processes," *Discrete Event Dyn. Syst.*, vol. 30, no. 1, pp. 81–124, Mar. 2020.

[12] Y. Qiao, M. C. Zhou, N. Q. Wu, and Q. H. Zhu, "Scheduling and control of startup process for single-arm cluster tools with residency time constraints," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1243–1256, Jul. 2017.

[13] Q. Zhu, M. Zhou, Y. Qiao, and N. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 3, pp. 389–400, Mar. 2018.

[14] N. Wu, M. Zhou, and Z. Li, "Short-term scheduling of crude-oil operations: Enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag.*, vol. 22, no. 2, pp. 64–76, Jun. 2015.

[15] F. Yang, N. Wu, Y. Qiao, M. Zhou, and Z. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.

[16] Z. Jia, J. Chen, and Y. Dai, "Decomposition and aggregation-based real-time analysis of assembly systems with geometric machines and small batch-based production tasks," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 988–999, Jul. 2021.

[17] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji, "Benchmarking protocols for evaluating small parts robotic assembly systems," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 883–889, Apr. 2020.

[18] F. Li, "Research on flexible scheduling problem based on bat algorithm and realization of customization production of flexible manufacturing," M.S. thesis, College Inf. Sci. Technol., Beijing Univ. Chem. Technol., Beijing, China, 2017.

[19] X. Wang, K. Xing, Y. Feng, and Y. Wu, "Scheduling of flexible manufacturing systems subject to no-wait constraints via Petri nets and heuristic search," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 10, pp. 6122–6133, Oct. 2021.

[20] Z. Li, N. Wu, and M. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets—A literature review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 437–462, Jul. 2012.

[21] Y. Chen, Z. Li, and M. Zhou, "Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 549–563, Apr. 2014.

[22] Y. Chen, Z. Li, and A. Al-Ahmari, "Nonpure Petri net supervisors for optimal deadlock control of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 2, pp. 252–265, Mar. 2013.

[23] Y. Chen and Z. Li, "On structural minimality of optimal supervisors for flexible manufacturing systems," *Automatica*, vol. 48, no. 10, pp. 2647–2656, Oct. 2012.

[24] H. Yue, K. Xing, H. Hu, W. Wu, and H. Su, "Resource failure and buffer space allocation control for automated manufacturing systems," *Inf. Sci.*, vol. 450, pp. 392–408, Jun. 2018.

[25] H. Liu, W. Wu, H. Su, and Z. Zhang, "Design of optimal Petri-net controllers for a class of flexible manufacturing systems with key resources," *Inf. Sci.*, vol. 363, pp. 221–234, Oct. 2016.

[26] Y. Huang, Y. Pan, and M. Zhou, "Computationally improved optimal deadlock control policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 2, pp. 404–415, Mar. 2012.

[27] Z. Li, G. Liu, M. Hanisch, and M. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.
[28] H. Hu, M. Zhou, Z. Li, and Y. Tang, "Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 109–121, Feb. 2013.
[29] H. Liu, K. Xing, W. Wu, M. Zhou, and H. Zou, "Deadlock prevention for flexible manufacturing systems via controllable siphon basis of Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 3, pp. 519–529, Mar. 2015.
[30] J. Luo, Z. Liu, and M. Zhou, "A Petri net based deadlock avoidance policy for flexible manufacturing systems with assembly operations and multiple resource acquisition," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3379–3387, Jun. 2019.
[31] G. Liu, L. Zhang, Y. Liu, Y. Chen, Z. Li, and N. Wu, "Robust deadlock control for automated manufacturing systems based on the max-controllability of siphons," *IEEE Access*, vol. 7, pp. 88579–88591, 2019.
[32] W. Duan, C. Zhong, X. Wang, A. U. Rehman, U. Umer, and N. Wu, "A deadlock prevention policy for flexible manufacturing systems modeled with Petri nets using structural analysis," *IEEE Access*, vol. 7, pp. 49362–49376, 2019.
[33] N. Du, H. Hu, and M. Zhou, "Robust deadlock avoidance and control of automated manufacturing systems with assembly operations using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1961–1975, Oct. 2020.
[34] J. Luo, Z. Liu, S. Wang, and K. Xing, "Robust deadlock avoidance policy for automated manufacturing system with multiple unreliable resources," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 3, pp. 812–821, May 2020.
[35] M. Bashir, J. Zhou, and B. B. Muhammad, "Optimal supervisory control for flexible manufacturing systems model with Petri nets: A place-transition control," *IEEE Access*, vol. 9, pp. 58566–58578, 2021.
[36] Y. Lu, Y. Chen, Z. Li, and N. Wu, "An efficient method of deadlock detection and recovery for flexible manufacturing systems by resource flow graphs," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1707–1718, Jul. 2022.
[37] X. Fan, H. Hu, B. Yang, Y. Liu, and G. He, "Event circuit structures for deadlock avoidance in flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 1, pp. 597–610, Jan. 2023.
[38] M. Teixeira, R. Malik, J. E. R. Cury, and M. H. de Queiroz, "Variable abstraction and approximations in supervisory control synthesis," in *Proc. Amer. Control Conf.*, Washington, DC, USA, Jun. 2013, pp. 132–137.
[39] M. Rosa, M. Teixeira, and R. Malik, "Exploiting approximations in supervisory control with distinguishers," *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 13–18, 2018.
[40] R. S. S. Aguiar, A. E. C. Cunha, J. E. R. Cury, and M. H. Queiroz, "Heuristic search of supervisors by approximated distinguishers," *IFAC Proc. Volumes*, vol. 46, no. 22, pp. 121–126, 2013.
[41] G. Bouzon, M. H. de Queiroz, and J. E. R. Cury, "Exploiting distinguishing sensors in supervisory control of DES," in *Proc. IEEE Int. Conf. Control Autom.*, Christchurch, New Zealand, Dec. 2009, pp. 442–447.
[42] M. Teixeira, J. E. R. Cury, and M. H. de Queiroz, "Exploiting distinguishers in local modular control of discrete-event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1431–1437, Jul. 2018.
[43] P. Mahdavinezhad, P. Gohari, and A. G. Aghdam, "Supervisory control of discrete-event systems with output: Application to hybrid systems," *Int. J. Control*, vol. 81, no. 9, pp. 1451–1461, Sep. 2008.
[44] T. Ushio and S. Takai, "Nonblocking supervisory control of discrete event systems modeled by Mealy automata with nondeterministic output functions," *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 799–804, Mar. 2016.
[45] X. Yin, "Supervisor synthesis for Mealy automata with output functions: A model transformation approach," *IEEE Trans. Autom. Control*, vol. 62, no. 5, pp. 2576–2581, May 2017.
[46] K. Jensen and L. M. Kristensen, *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Berlin, Germany: Springer, 2009.
[47] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.
[48] M. Teixeira, R. Malik, J. E. R. Cury, and M. H. de Queiroz, "Supervisory control of DES with extended finite-state machines and variable abstraction," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 118–129, Jan. 2015.
[49] C. Ma and W. M. Wonham, "Nonblocking supervisory control of state tree structures," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 782–793, May 2006.
[50] T. Jiao and Y. Bai, "On efficient relabeling of symmetric discrete-event systems," *Asian J. Control*, vol. 24, no. 6, pp. 3603–3614, Nov. 2022.

TING JIAO (Member, IEEE) received the B.Eng. degree in automation from Central South University, Changsha, China, in 2010, and the M.Eng. degree in control science and engineering and the Ph.D. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2013 and 2017, respectively. He was an International Visiting Ph.D. Student with the University of Toronto, Toronto, ON, Canada, from 2014 to 2015. He was a Visiting Scholar with Osaka City University, Osaka, Japan, from 2022 to 2023. He is currently an Associate Professor with the School of Automation and Software Engineering, Shanxi University, Taiyuan, China. His current research interests include supervisory control and the reconfiguration of discrete-event systems.

HONGBING SHI received the B.Eng. degree in automation from Shanxi University, Taiyuan, China, in 2019, where he is currently pursuing the master's degree in control science and engineering. His research interests include the supervisory control of discrete-event systems and the optimal control of the thermal power generation.