Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Decentralized Control of Discrete Event Systems Using Prioritized Composition with Exclusion

Wenbin Qiu, Ratnesh Kumar
Department of ECE
Iowa State University
Ames, IA 50011

Vigyan Chandra
Department of Technology
Eastern Kentucky University
Richmond, KY 40475

***Abstract*— This paper studies the decentralized control problem of discrete event systems via prioritized composition with exclusion (PCX). PCX was obtained by extending prioritized synchronous composition (PSC) in order to model various Boolean modes of decision fusion resulting from the exclusivity of participation in system interactions [1]. PCX decision fusion allows additional modes of decision fusion beyond the AND/OR architecture considered in [13]. In this paper, we present a definition of PCX co-observability, which together with controllability and $L_m(G)$-closure serves as a necessary and sufficient condition for the existence of decentralized supervisors under PCX decision fusion. We also provide an algorithm for testing the PCX co-observability which has the computational complexity that is quadratic in plant states and cubic in specification states. Next the properties of PCX fusion architecture are presented. Finally, the exclusive operation is extended from two supervisors to n supervisors, and corresponding decentralized control has been studied.**

Keywords: Discrete event system, Decentralized supervisory control, Prioritized composition with exclusion, PCX co-observability

I. Introduction

Supervisory control of discrete event systems (DESs) was first proposed by Ramadge and Wonham [9]. Later in [7], [6], Lin and Wonham considered the supervisory control with partial observation and introduced the notion of *observability*. When more than one supervisors control a system, called decentralized control, the condition of *co-observability* introduced by Cieslak *et al.* [2] and Rudie and Wonham [11] plays a key role. Control decisions from local supervisors were fused using *conjunctive* fusion rule in which an event is globally enabled if and only if it is enabled by all local supervisors. A non-conjunctive fusion rule based decentralized control was first introduced by Prosser *et al.* [8]. Yoo and Lafortune developed a more complete theory and proposed a general decision fusion architecture based on the *conjunctive* (AND)/*disjunctive* (OR) fusion rules [13]. In the AND/OR architecture, controllable events are partitioned into two disjoint sets: global enablement decision for one is made using the conjunctive fusion rule, and for the other using the disjunctive fusion rule.

This work was supported in part by the National Science Foundation under the grants NSF-ECS-0218207, NSF-ECS-0244732 and NSF-EPNES-0323379, and a DoD-EPSCoR grant from the Office of Naval Research under the grant N000140110621.

The supervisory control of DESs and the interaction of various local supervisors can be achieved through appropriate composition of their automata representations. In the conventional supervisory control, the interaction of a plant and a supervisor is obtained through *strict synchronous composition* (SSC), where the plant and the supervisor have the same event set. Since feasible uncontrollable events cannot be disabled by a supervisor, the supervisor is required to synchronously execute all the uncontrollable events that the plant can execute.

To relax such synchronization requirement, Heymann [3] introduced *prioritized synchronous composition* (PSC), where a priority set is associated with each system. For an event to be enabled in the interconnected system, it must be enabled in all systems whose priority sets contain that event. In the setting of supervisory control, the priority set of the supervisor is chosen to be the controllable event set, and the supervisor is not required to participate in the occurrence of uncontrollable events. Supervisory control of DESs via PSC has been studied in [12], [4], [5].

As discussed in Chandra *et al.* [1], PSC of two systems P and Q allows for four modes of Boolean interaction, namely, "P AND Q", "P OR Q", "ONLY P", and "ONLY Q". While this is adequate to model the interaction of plant and supervisor, it can not model the interaction of local supervisors which can have additional modes of interaction based upon exclusive operations. For example, if there are two local supervisors P and Q, the additional possible modes of interactions include "exclusive P", "exclusive Q", "exclusive P or exclusive Q", and "exclusive P and exclusive Q". To model the exclusivity of participation, a generalization of PSC, *prioritized composition with exclusion* (PCX), was proposed in [1] by introducing an exclusivity set beyond the two priority sets. For enabling an event in the exclusivity set, it needs to be enabled by exactly one supervisor.

PCX based decision fusion extends the AND/OR architecture considered in [8], [13]. There are several reasons for why one would want to use a PCX based decentralized control, such as:

1) PCX allows for all possible modes of decision fusion that are enablement-based, and hence give rise to a general memoryless control decision fusion architecture.

2) Like PSC, PCX does not require the participation of local supervisions in occurrence of uncontrollable events.
3) Control problem at hand may necessitate that certain decisions be exclusive (see the motivating example in section 3.1), and so there is no choice but to use PCX.

In this paper, we study the decentralized control using PCX as a mechanism for interaction among supervisors. We first discuss the default decisions and the fusion rules in Section II. The default decisions define the control actions when supervisors cannot make unambiguous decision. Next, we give definitions of PCX co-observability for two systems. The notion of PCX co-observability serves as a condition for deciding whether a ceratin language can be achieved by local supervisors that are combined using the PCX composition. We establish a necessary and sufficient condition for the existence of PCX supervisors. We present an algorithm for testing the PCX co-observability. The algorithm is quadratic in plant states and cubic in specification states, and matches the complexity for decentralized control under AND/OR fusion architecture. The properties of PCX fusion architecture are presented in Section IV, in particular we study the relationship between the AND/OR architecture and the PCX architecture. In Section V, extension is presented for PCX operations among n systems, where to define PCX a total of 2^n-n-1 exclusivity sets are required. Section VI summarizes this paper.

II. Decentralized Control using PSC and PCX

The prioritized synchronous composition (PSC) was first introduced by Heymann in [3]. In PSC, a priority set is associated with each interacting system. For an event to be enabled in the interconnected system, it must be enabled in all systems whose priority sets contain that event. For a system with two components, PSC is formally defined as follows:

Definition 1: Given two systems $S_1 = (Q_1, \Sigma, \delta_1, q_1^0, Q_1^m)$ and $S_2 = (Q_2, \Sigma, \delta_2, q_2^0, Q_2^m)$ with a common event set Σ, let $A_1, A_2 \subseteq \Sigma$ denote the priority event sets of S_1 and S_2 respectively. The PSC of S_1 and S_2 is denoted by $S_{1\,A_1}\|_{A_2} S_2 := (Q, \Sigma, \delta, q^0, Q^m)$, where $Q := Q_1 \times Q_2$, $q^0 := (q_1^0, q_2^0)$, $Q^m := Q_1^m \times Q_2^m$, and the transition function $\delta : Q \times \Sigma \to Q$ is defined as follows: $\forall q = (q_1, q_2) \in Q, \sigma \in \Sigma, \delta(q, \sigma) :=$

$$\begin{cases} \delta_1(q_1,\sigma), \delta_2(q_2,\sigma), & \text{if } \delta_1(q_1,\sigma), \delta_2(q_2,\sigma) \text{ defined;} \\ \delta_1(q_1,\sigma), q_2), & \text{if } \delta_1(q_1,\sigma) \text{ defined,} \\ & \delta_2(q_2,\sigma) \text{ not defined, } \sigma \notin A_2; \\ (q_1, \delta_2(q_2,\sigma)), & \text{if } \delta_1(q_1,\sigma) \text{ not defined,} \\ & \delta_2(q_2,\sigma) \text{ defined, } \sigma \notin A_1; \\ \text{undefined} & \text{otherwise.} \end{cases}$$

In order to allow additional enablement-based modes of event enablement, a more general composition operation, prioritized composition with exclusion (PCX), was introduced by Chandra *et al.* [1]. ("Enablement-based" means whenever an event is globally enabled, it is the case that not all interacting systems disable it. Extension of PCX to allow for "non-enablement-based" decision fusion was also introduced in [1], called prioritized composition with exclusion and generation, PCXG.) In PCX, an exclusivity set is introduced to model the exclusivity of participation in event enablement. An event in the exclusivity set can be enabled if and only if exactly one system enables it. For two interacting systems, PCX is formally defined as follows:

Definition 2: Given two systems $S_1 = (Q_1, \Sigma, \delta_1, q_1^0, Q_1^m)$ and $S_2 = (Q_2, \Sigma, \delta_2, q_2^0, Q_2^m)$ with a common event set Σ, let $A_1, A_2 \subseteq \Sigma$ be the priority event set of S_1 and S_2 respectively, and X be the exclusivity set. The prioritized composition with exclusion of S_1 and S_2 is denoted by $S_1 \;{}_{A_1}\overset{X}{\|}_{A_2}\; S_2 := (Q, \Sigma, \delta, q^0, Q^m)$, where $Q := Q_1 \times Q_2$, $q^0 := (q_1^0, q_2^0)$, $Q^m := Q_1^m \times Q_2^m$, and the transition function $\delta : Q \times \Sigma \to Q$ is defined as follows: $\forall q = (q_1, q_2) \in Q, \sigma \in \Sigma, \delta(q, \sigma) :=$

$$\begin{cases} (\delta_1(q_1,\sigma), \delta_2(q_2,\sigma)) & \text{if } \delta_1(q_1,\sigma), \delta_2(q_2,\sigma) \text{ defined,} \\ & \sigma \notin X; \\ (\delta_1(q_1,\sigma), q_2) & \text{if } \delta_1(q_1,\sigma) \text{ defined,} \\ & \delta_2(q_2,\sigma) \text{ not defined, } \sigma \notin A_2; \\ (q_1, \delta_2(q_2,\sigma)) & \text{if } \delta_1(q_1,\sigma) \text{ not defined,} \\ & \delta_2(q_2,\sigma) \text{ defined, } \sigma \notin A_1; \\ \text{undefined} & \text{otherwise;} \end{cases}$$

We apply both PSC and PCX for the decentralized supervisory control of discrete event systems. PSC is used to model the interaction of plant and supervisor, where the priority sets for plant and supervisor are Σ (set of plant events) and Σ_c (set of controllable events), respectively. PCX, on the other hand, is used to fuse the control decisions of local supervisors to obtain global control decision. The fused PCX supervisor is called S_{PCX}. Then the controlled system can be expressed as follows: $S_{PCX}/G = PSC(G, S_{PCX}) = G_{\Sigma}\|_{\Sigma_C} S_{PCX}$.

Assume for simplicity of presentation that there are two local supervisors S_1 and S_2, i.e., the supervisor index is $I = \{1, 2\}$. The more general case of arbitrary number of supervisors is presented at the end of the paper. For ith local supervisor, its priority event set is $A_i (i \in I)$. Since a supervisor needs to control events in its priority set, we assume for each $i \in I$, $A_i \subseteq \Sigma_{ci}$, where $\Sigma_{ci} \subseteq \Sigma_c$ is the set of controllable events for the ith supervisor with $\cup_i \Sigma_{ci} = \Sigma_c$. Let $A = \underset{i}{\cup} A_i$ denote the set of all priority events, and X the exclusivity set. Then the PCX supervisor is expressed as: $S_{PCX} = PCX(S_1, S_2) = S_1 \;{}_{A_1}\overset{X}{\|}_{A_2}\; S_2$.

With two priority event sets A_1 and A_2, and one exclusivity event set X, the controllable event set Σ_c is partitioned into eight subsets: $\Sigma_\wedge = (A_1 \cap A_2) - X$, $\Sigma_\vee = \Sigma_c - (A_1 \cup A_2 \cup X)$, $\Sigma_{A_1} = (A_1 - A_2) - X$, $\Sigma_{A_2} = (A_2 - A_1) - X$, $\Sigma_{XA_1} = (A_1 - A_2) \cap X$, $\Sigma_{XA_2} = (A_2 - A_1) \cap X$, $\Sigma_\oplus = X - (A_1 \cap A_2)$ and $\Sigma_\emptyset = (A_1 \cap A_2) \cap X$. The way local decisions are fused to obtain a global decision depends on the fusion rules for the event in question. Let $A = \cup_i A_i$ denote set of all events that belong to at least one priority set. For an event σ in partition $A-X$, it is enabled if it is enabled by all local supervisors whose priority sets contain σ (*conjunctive rule*). For $\sigma \in \Sigma_\vee := \Sigma - (A \cup X)$, it is enabled if it is enabled by at least one local supervisor (*disjunctive rule*). For $\sigma \in \Sigma_\oplus := X - A$, it is enabled if it is enabled by exactly one local supervisor (*exclusive rule*). For $\sigma \in \Sigma_{XA_i} := X \cap A_i$, σ is enabled if it is enabled by the only local supervisor whose priority set contains σ (*exclusive rule with priority*).

A. Default Decision Rule

In order to take a control action (enable/disable), a supervisor needs to be unambiguous about it. For this if an event needs to be enabled (resp., disabled) following a trace s, then that event must also be enabled (reps., disabled) following an indistinguishable trace $t \in M_i^{-1}M_i(s)$ that is in the legal behavior. (Here M_i is the observation mask of the ith supervisor.)

A default decision rule is to be used when a supervisor cannot make decision unambiguously. The default decision rule for PCX architecture is stated as follows: *Whenever a supervisor is ambiguous, the default decision is to enable an event if and only if the event is in the priority set of that supervisor.* Table I shows default decisions and fusion rules for all sets in PCX partition for two local supervisors. "S_1" and "S_2" indicate the decisions by the two local supervisors. "S_{PCX}" indicates the fused decision. This table is also suitable for PSC if we only consider the sets $\Sigma_\vee$, $\Sigma_\wedge$, Σ_{A_1}, and Σ_{A_2}.

	Default Decision	Fusion Rule
$\Sigma_\vee$	$S_1=0$ $S_2=0$	$S_1=0$ and $S_2=0 \Leftrightarrow S_{PCX}=0$ $S_1=1$ or $S_2=1 \Leftrightarrow S_{PCX}=1$
$\Sigma_\wedge$	$S_1=1$ $S_2=1$	$S_1=0$ or $S_2=0 \Leftrightarrow S_{PCX}=0$ $S_1=1$ and $S_2=1 \Leftrightarrow S_{PCX}=1$
Σ_{A_1}	$S_1=1$ $S_2=0$	$S_1=0 \Leftrightarrow S_{PCX}=0$ $S_1=1 \Leftrightarrow S_{PCX}=1$
Σ_{A_2}	$S_1=0$ $S_2=1$	$S_2=0 \Leftrightarrow S_{PCX}=0$ $S_2=1 \Leftrightarrow S_{PCX}=1$
Σ_{XA_1}	$S_1=1$ $S_2=0$	$S_1=0$ or $S_2=1 \Leftrightarrow S_{PCX}=0$ $S_1=1$ and $S_2=0 \Leftrightarrow S_{PCX}=1$
Σ_{XA_2}	$S_1=0$ $S_2=1$	$S_1=1$ or $S_2=0 \Leftrightarrow S_{PCX}=0$ $S_1=0$ and $S_2=1 \Leftrightarrow S_{PCX}=1$
$\Sigma_\emptyset$	$S_1=1$ $S_2=1$	$S_{PCX}\equiv 0$
$\Sigma_\oplus$	$S_1=0$ $S_2=0$	($S_1=0$, $S_2=0$) or ($S_1=1$, $S_2=1$) $\Leftrightarrow S_{PCX}=0$ ($S_1=1$, $S_2=0$) or ($S_1=0$, $S_2=1$) $\Leftrightarrow S_{PCX}=1$

TABLE I

DEFAULT DECISIONS AND FUSION RULES FOR PCX (1: ENABLE; 0: DISABLE)

B. Existence Conditions of PCX Supervisors

For notational convenience, we define $I_c(\sigma) := \{i \in I | \sigma \in \Sigma_{ci}\}$ and $I_A(\sigma) := \{i \in I | \sigma \in A_i\}$ to denote the index set of all supervisors whose controllable event set contains σ, and the index set of all supervisors whose priority event set contains σ, respectively. Then, the notion of PCX co-observability is defined as follows.

Definition 3: A language $K \subseteq L(G) = \overline{L(G)}$ is said to be $\{A_i, X, \Sigma_{ci}, M_i\}$ PCX co-observable $(i = 1, 2)$ w.r.t. $L(G)$ if

Conjunct 1: $\forall s \in \overline{K}, \sigma \in \Sigma_c - (A \cup X) = \Sigma_\vee, s\sigma \in \overline{K}, \exists i \in I_c(\sigma)\ s.t.\ [M_i^{-1}M_i(s) \cap \overline{K}]\sigma \cap L(G) \subseteq \overline{K}$.

Conjunct 2: $\forall s \in \overline{K}, \sigma \in A - \Sigma_\emptyset, s\sigma \in L(G) - \overline{K}, \exists i \in I_A(\sigma)\ s.t.\ [M_i^{-1}M_i(s)]\sigma \cap \overline{K} = \emptyset$.

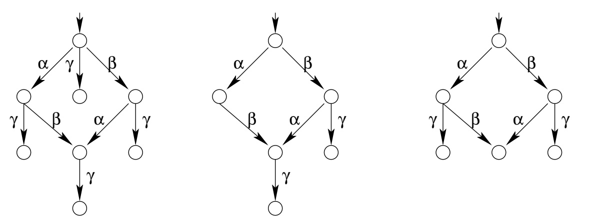
Conjunct 3: $\forall s \in \overline{K}, \sigma \in \Sigma_\emptyset : s\sigma \notin \overline{K}$.

Conjunct 4: $\forall s \in \overline{K}, \sigma \in X - A = \Sigma_\oplus, s\sigma \in \overline{K}, \exists$ exactly one $i \in I_c(\sigma)\ s.t.\ [M_i^{-1}M_i(s) \cap \overline{K}]\sigma \cap L(G) \subseteq \overline{K}$.

Conjunct 1 indicates that for an event in $\Sigma_\vee = \Sigma_c - A$ to be enabled it needs to be unambiguously enabled by at least one supervisor that controls the event. Conjunct 2 indicates that for an event in A to be disabled it needs to be unambiguously disabled by at least one supervisor whose priority set contains that event. Conjunct 3 indicates that events in $\Sigma_\emptyset$ will be disabled regardless of decisions by local supervisors. Conjunct 4 indicates that if an event in $\Sigma_\oplus$ needs to be enabled, there exists one and only one supervisor which can unambiguously enable that event.

The following example illustrates the definition of PCX co-observability.

Example 1: Figure 1 shows a plant model G, and two specification models H and H'. Let $M_i : \Sigma \to \Sigma_{oi}$ be projection type observation masks. $\Sigma_{o1} = \{\alpha, \gamma\}$, $\Sigma_{o2} = \{\beta, \gamma\}$, $\Sigma_{c1} = \Sigma_{c2} = \{\gamma\}$. The priority sets for both supervisors are empty $A_1 = A_2 = \emptyset$ and the exclusivity set is $X = \{\gamma\}$. Therefore, $\Sigma_\oplus = \{\gamma\}$ and $\Sigma_{A_1} = \Sigma_{A_2} = \Sigma_{XA_1} = \Sigma_{XA_2} = \Sigma_\wedge = \Sigma_\vee = \Sigma_\emptyset = \emptyset$. Then it can be verified that $L(H)$ is $\{A_i, X, \Sigma_{ci}, M_i\}$ PCX co-observable w.r.t. $L(G)$, whereas $L(H')$ is not. ■



(a) Plant G (b) Specification H (c) Specification H'

Fig. 1. Diagram illustrating Example 1

Based on the definition of PCX co-observability, we present a necessary and sufficient condition for the existence of decentralized PCX-based supervisors as follows.

Theorem 1: Given two supervisors $(I = \{1, 2\})$, consider a language $K \subseteq L_m(G) (K \neq \emptyset)$ and a fixed partition of Σ_c such that $\Sigma_c = A \cup X \cup \Sigma_\vee$, where A is the overall priority set $A := \bigcup_{i \in I} A_i$ (A_i is the priority set of supervisor i), X is the exclusivity set, and $\Sigma_\vee = \Sigma_c - (A \cup X)$. Assume the controlled system is the PSC composition of a plant and a fused PCX supervisor. There exists a nonblocking PCX supervisor S_{PCX} such that $L_m(S_{PCX}/G) = K$ if and only if the three following conditions hold:

1) K is controllable w.r.t. $L(G)$ and $\Sigma_u := \Sigma - \Sigma_c$ ($\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$);
2) K is $\{A_i, X, \Sigma_{ci}, M_i\}$ PCX co-observable $(i \in I)$ w.r.t. $L(G)$;
3) K is $L_m(G)$-closed ($\overline{K} \cap L_m(G) = K$).

III. Algorithm for Testing PCX Co-observability

For testing the PCX co-observability, we develop an algorithmic test as in [10], [13], where a special automaton is constructed to mark the bad traces which violate the co-observability condition. The only marked state in the special automaton is the so called "dump" state.

We do not present all the details of the testing automata in this paper. Essentially, the testing automaton has a transition function so that for all $s, s_1, s_2 \in \overline{K}$, it tracks traces satisfying the conditions $[M_1(s) = M_1(s_1)] \wedge [M_2(s) = M_2(s_2)]$(see for example [10], [13]). Here, we only discuss the violating condition for each partition set and the corresponding state space in each testing automaton. This is shown in Table II, where Q^G is the state set of the plant automaton G, Q^H is the state set of the specification automaton H, and d is the dump state.

Partition Set	Violating Conditions	State Space Required
$\sigma \in \Sigma_\vee$	$[s_1\sigma \in L(G) - L(H)]$ $\wedge[s_2\sigma \in L(G) - L(H)]$ $\wedge[s\sigma \in L(H)]$	$(Q^G \times Q^H \times Q^G \times Q^H \times Q^H) \cup \{d\}$
$\sigma \in \Sigma_\wedge$	$[s_1\sigma \in L(H)]$ $\wedge[s_2\sigma \in L(H)]$ $\wedge[s\sigma \in L(G) - L(H)]$	$(Q^H \times Q^H \times Q^G \times Q^H)$ $\cup\{d\}$
$\sigma \in \Sigma_{A_1}$	$[s_1\sigma \in L(H)]$ $\wedge[s\sigma \in L(G) - L(H)]$	$(Q^H \times Q^G \times Q^H) \cup \{d\}$
$\sigma \in \Sigma_{A_2}$	$[s_2\sigma \in L(H)]$ $\wedge[s\sigma \in L(G) - L(H)]$	$(Q^H \times Q^G \times Q^H) \cup \{d\}$
$\sigma \in \Sigma_{XA_1}$	$[s_1\sigma \in L(H)]$ $\wedge[s_2\sigma \in L(G) - L(H)]$ $\wedge[s\sigma \in L(G) - L(H)]$	$(Q^H \times Q^G \times Q^H \times Q^G \times Q^H) \cup \{d\}$
$\sigma \in \Sigma_{XA_2}$	$[s_1\sigma \in L(G) - L(H)]$ $\wedge[s_2\sigma \in L(H)]$ $\wedge[s\sigma \in L(G) - L(H)]$	$(Q^G \times Q^H \times Q^H \times Q^G \times Q^H) \cup \{d\}$
$\sigma \in \Sigma_\oplus$	$[s_1\sigma \in L(G) - L(H)]$ $\wedge[s_2\sigma \in L(G) - L(H)]$ $\wedge[s\sigma \in L(H)]$ or: $[s_1\sigma \in L(H)]$ $\wedge[s_2\sigma \in L(H)]$ $\wedge[s\sigma \in L(H)]$	$(Q^G \times Q^H \times Q^G \times Q^H \times Q^H) \cup \{d\}$
$\sigma \in \Sigma_\emptyset$	$s\sigma \in L(H)$	$Q^H \cup \{d\}$

TABLE II

Violating conditions and state spaces

Each violating condition causes a transition to the dump state. In each condition, we assume that the event σ in trace $s_i\sigma$ $(i \in \{1,2\})$ can be controlled by the corresponding supervisor S_i $(i \in \{1,2\})$. If σ is not controlled by a supervisor i, we omit condition on $s_i\sigma$ $(i \in \{1,2\})$. For example, if $\sigma \in \Sigma_\vee$ and $\sigma \in \Sigma_{c1} - \Sigma_{c2}$, we only check if the condition $[s_1\sigma \in L(G) - L(H)] \wedge [s\sigma \in L(H)]$ is violated or not, ignoring the condition $s_2\sigma \in L(G) - L(H)$ in the row for $\Sigma_\vee$ in the above table.

Assume the testing automata for events in $\Sigma_\wedge$, $\Sigma_\vee$, Σ_{A_1}, Σ_{A_2}, Σ_{XA_1}, Σ_{XA_2}, $\Sigma_\oplus$, and $\Sigma_\emptyset$ are $T_i(G,H)$ $(i \in I_{PCX} = \{1,2,3,4,5,6,7,8\})$, respectively. In construction of $T_i(G,H)$, the maximum that is needed is two copies of plant automata and three copies of specification automata. Therefore, the algorithm for testing PCX co-observability is quadratic in plant states and cubic in specification states and has the same computational complexity as the algorithm for D&A co-observability presented in [13]. So we have the following result:

Theorem 2: Given two deterministic automata G and H, $K = L_m(H)$ is PCX co-observable if and only if $L_m(T_i(G,H)) = \emptyset$ for all $i \in I_{PCX}$, i.e., if and only if the dump state d is not reachable in all testing automata $T_i(G,H)$. The complexity of constructing $T_i(G,H)$ is $O(|Q^G|^2|Q^H|^3)$, which is also the complexity of testing PCX co-observability.

IV. Properties of PCX Fusion Architecture

In this section, we discuss the properties of PCX fusion architecture.

Property 1: In a PCX partition, if all local supervisors share the same priority set $A_1 = A_2 \subseteq \Sigma_c$ and the exclusivity set is empty $X = \emptyset$, then the PCX architecture is reduced to AND/OR architecture.

Property 2: In a PCX partition, if Σ_c is the priority set for all local supervisors $A_1 = A_2 = \Sigma_c$ and the exclusivity set is empty $X = \emptyset$, then the PCX architecture is reduced to a conjunctive (AND) architecture.

Property 3: In a PCX partition, if all the priority sets and the exclusivity set are empty $A_1 = A_2 = X = \emptyset$, then the PCX architecture is reduced to a disjunctive (OR) architecture.

V. PCX for Multiple Supervisors

In the previous discussion, the PCX operation with two local supervisors was explored. When more than two supervisors interact, more (than one) exclusivity sets are needed. For n supervisors, we can define the exclusivity set as follows:

Definition 4: Given n supervisors with the index set $I = \{1, 2, \cdots, n\}$. $X_E \subseteq \Sigma$ is called the exclusivity set over the index set $E \subset I$ of supervisors, where $2 \leq |E| \leq n$. For an event in X_E to be enabled, it must be enabled by exactly one supervisor in E.

Since the exclusive operation should be defined over more than two supervisors, we require $|E| \geq 2$. For n supervisors, there can be $m = 2^n - n - 1$ different exclusivity sets. Let X denote the overall exclusivity set: $X := \bigcup_{j=1\ldots m} X_{E_j}$. For any two exclusivity sets, we have the property stated as follows:

Property 4: Given n supervisors with the index set $I = \{1, 2, \cdots, n\}$. X_E and X_F are two exclusivity sets over E and F respectively ($E, F \subseteq I, 2 \le |E| \le n$ and $2 \le |F| \le n$). Then $X_E \supseteq X_F$ if and only if $E \subseteq F$.

Property 4 can be derived from the definition of exclusivity set. Intuitively, it may be paraphrased as: "the more events the exclusivity set includes, the less supervisors it should involve, and vice versa". For the convenience of further discussion, we define an exclusivity index set $I_X(\sigma) := \bigcup_{\sigma \in X_{E_j}} E_j \quad (1 \le j \le m,\ m = 2^n - n - 1)$, which denotes the index set of all supervisors which are involved in exclusive operations of events σ by way of membership in the exclusivity sets X_{E_j}.

Then we can define the extended prioritized composition with exclusion as follows:

Definition 5: Given n systems $S_i = (Q_i, \Sigma_i, \delta_i, q_i^0, Q_i^m)$ $(i \in I = \{1, 2, \cdots, n\}$, let $A_i \subseteq \Sigma$ be the priority event set of S_i and X_{E_j} $(1 \le j \le m, m = 2^n - n - 1)$ be the exclusivity set over the index set E_j ($E_j \subseteq I$ and $2 \le |E_j| \le n$). The prioritized composition with exclusion of S_i $(i \in I)$ is denoted by $PCX(S_1, \cdots, S_n) := (Q, \Sigma, \delta, q^0, Q^m)$, where $Q := Q_1 \times \cdots \times Q_n, \Sigma := \bigcup_{i \in I} \Sigma_i, q^0 := (q_1^0, \cdots, q_n^0), Q^m := Q_1^m \times \cdots \times Q_n^m$. For $q = (q_1, \cdots, q_n) \in Q$, let $I_e(q, \sigma)$ and $I_d(q, \sigma)$ denote the enabled and disabled index set at the state q for $\sigma \in \Sigma$ as: $I_e(q, \sigma) := \{i \in I | \delta_i(q_i, \sigma)$ defined$\}$, and $I_d(q, \sigma) := \{i \in I | \delta_i(q_i, \sigma)$ not defined$\}$. We let $q_i{}' = \delta_i(q_i, \sigma)$ for all $i \in I_e(q, \sigma)$, and $q_i{}' = q_i$ otherwise. The transition function $\delta : Q \times \Sigma \to Q$ is defined as: $\delta(q, \sigma) :=$

$$\begin{cases} q' = (q_1', \cdots, q_n') & \text{if } |I_e(q,\sigma) \cap I_X(\sigma)| \le 1 \\ & \text{and } I_d(q,\sigma) \cap I_A(\sigma) = \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$

With the definition of PCX for n systems, we can define the PCX co-observability for n supervisors as follows:

Definition 6: Let $I = \{1, 2, \cdots, n\}$ and $m = 2^n - n - 1$. A language $K \subseteq L(G) = \overline{L(G)}$ is said to be $\{A_i, X_j, \Sigma_{ci}, M_i\}$ PCX co-observable $(i \in I, j \in \{1, \cdots, m\})$ w.r.t. $L(G)$, if

Conjunct 1: $\forall s \in \overline{K}, \sigma \in \Sigma_c - (A \cup X) = \Sigma_\vee, s\sigma \in \overline{K}, \exists i \in I_c(\sigma), s.t.\ [M_i^{-1} M_i(s) \cap \overline{K}]\sigma \cap L(G) \subseteq \overline{K}$.

Conjunct 2: $\forall s \in \overline{K}, \sigma \in A - X, s\sigma \in L(G) - \overline{K}, \exists i \in I_A(\sigma), s.t.\ [M_i^{-1} M_i(s)]\sigma \cap \overline{K} = \emptyset$.

Conjunct 3: $\forall s \in \overline{K}, \sigma \in \Sigma_\emptyset = \{\sigma \in A \cap X | \, |I_A(\sigma)| > 1\} : s\sigma \notin \overline{K}$.

Conjunct 4: $\forall s \in \overline{K}$, $\sigma \in X - \Sigma_\emptyset$, $s\sigma \in \overline{K}$, $\forall j \in \{1, ..., m\}$, $\exists$ exactly one $i \in E_j$ with $\sigma \in X_{E_j}$ $s.t.\ [M_i^{-1} M_i(s) \cap \overline{K}]\sigma \cap L(G) \subseteq \overline{K}$.

Based on Definition 6, we give a necessary and sufficient condition as in Theorem 1 for decentralized control using n PCX-based supervisors.

Theorem 3: Let $I = \{1, 2, \cdots, n\}$ and $m = 2^n - n - 1$. Consider a language $K \subseteq L_m(G)$ $(K \ne \emptyset)$ and a fixed partition of Σ_c such that $\Sigma_c = A \cup X \cup \Sigma_\vee$, where A is the union of all priority sets $A := \bigcup_{i \in I} A_i$ (A_i is the priority set of supervisor i), X is the overall exclusivity set, $X := \bigcup_{j=1 \dots m} X_j$, and $\Sigma_\vee = \Sigma_c - (A \cup X)$. Assume the controlled system is the PSC composition of a plant and a PCX-fused set of supervisors. There exists a nonblocking PCX-fused supervisor S_{PCX} such that $L_m(S_{PCX}/G) = K$ if and only if the three following conditions hold:

1) K is controllable w.r.t. $L(G)$ and Σ_{uc};
2) K is $\{A_i, X_j, \Sigma_{ci}, M_i\}$ PCX co-observable $(i \in I, j \in \{1, \cdots, m\})$ w.r.t. $L(G)$;
3) K is $L_m(G)$-closed.

VI. Conclusion

In this paper the decentralized control of discrete event systems via PCX is studied. A necessary and sufficient condition for the existence of PCX-based supervisors is presented based on the definition of PCX co-observability. A polynomial algorithm for testing the PCX co-observability is provided. Finally, extension is presented for the PCX based decision fusion from 2 supervisors to n supervisors.

References

[1] V. Chandra, Z. Hunag, W. Qiu, and R. Kumar. Prioritized composition with exclusion and generation for the interaction and control of discrete event systems. *Mathematical and Computer Modeling of Dynamical Systems*, 9(3):255–280, 2003.

[2] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya. Supervisory control of discrete event processes with partial observation. *IEEE Transactions on Automatic Control*, 33(3):249–260, 1988.

[3] M. Heymann. Concurrency and discrete event control. *IEEE Control Systems Magazine*, 10(4):103–112, 1990.

[4] R. Kumar and M. Heymann. Masked prioritized synchronization for interaction and control of discrete event systems. *IEEE Transactions on Automatic Control*, 45(11):1970–1982, 2000.

[5] R. Kumar and M. A. Shayman. Non-blocking supervisory control of nondeterministic systems via prioritized synchronization. *IEEE Transactions on Automatic Control*, 41(8):1160–1175, August 1996.

[6] F. Lin and W. M. Wonham. Decentralized supervisory control of discrete event systems. *Information Sciences*, 44:199–224, 1988.

[7] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173–198, 1988.

[8] J. H. Prosser and M. Kam amd H. G. Kwatny. Decision fusion and supervisor synthesis in decentralized discrete-event systems. In *Proceedings of 1997 American Control Conference*, pages 2251–2255, 1997.

[9] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.

[10] K. Rudie and J. C. Willems. The computational complexity of decentralized discrete-event control problems. *IEEE Transactions on Automatic Control*, 40(7):1313–1319, July 1995.

[11] K. Rudie and W. M. Wonham. Think globally, act locally: decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, November 1992.

[12] M. Shayman and R. Kumar. Supervisory control of nondeterministic systems with driven events via prioritized synchronization and trajectory models. *SIAM Journal of Control and Optimization*, 33(2):469–497, March 1995.

[13] T. S. Yoo and S. Lafortune. A general architecture for decentralized supervisory control of discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 12(3):335–377, July 2002.