Prioritized Composition with Exclusion and Generation for the Interaction and Control of Discrete Event Systems

V. CHANDRA*, Z. HUANG†, W. QIU‡, R. KUMAR §

ABSTRACT

Interaction of multiple discrete event systems (DESs) represented as automata are carried out using composition operations. These operations on automata enforce concurrency, wherein an event exists in the composed automaton if it exists in the participating states of the interacting automata possessing the event in their event set. Heymann generalized this by introducing event priorities, wherein an event exists in the composed automaton if it exists in the participating states of the interacting automata having priority over the event. For two interacting automata P and Q, while prioritized composition can model the $P \land Q$, $P \lor Q$, and $P \land Q$ boolean interactions, it cannot model boolean interactions which require exclusivity of participation, namely, “exclusive P”, “exclusive Q”, “exclusive P or exclusive Q”, “exclusive P and exclusive Q”. In order to also model these additional interactions we propose a generalization of prioritized composition by introducing an exclusivity set besides the existing priority sets. The resulting composition is called prioritized composition with exclusion. We also introduce prioritized composition with exclusion and generation that allows for all sixteen boolean modes of interaction possible when two automata interact. This is done by the further introduction of a nor-generative set. This event set together with the two priority sets and an exclusivity set makes it possible to model eight additional boolean interactions which do not require either of the interacting automata to participate for the event to be enabled in the composed automaton. The applicability of these interactions to decentralized supervisory decision fusion and in composing the rules based model of systems has been illustrated.

Key words: discrete event systems, synchronous composition, prioritized synchronous composition, prioritized composition with exclusion, generalized composition, supervisory control, rules based model, decentralized control.

*Department of Technology, Eastern Kentucky University, Richmond, KY, 40475, USA.
†Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, 40506, USA.
‡Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50011, USA.
§Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50011, USA. Email: rkumar@iastate.edu
1 INTRODUCTION

A system usually consists of several interacting subsystems. The restriction or control of the behavior of such a system can be achieved through (i) restriction or control of behavior of the individual subsystems, and (ii) restriction or control of the interactions among the subsystems. Much work has been done on the restriction/control of (sub)systems [4, 11]. In this paper we examine and propose ways of restricting/controlling interactions among the subsystems.

Interaction of discrete event systems (DESs) represented as automata is carried out using composition operations [9, 15]. Restriction on such interactions can be obtained by restrictions such as requiring the synchronization of shared events, an interaction known as synchronous composition (SC). Under SC an event exists in the composed automaton if it exists in the participating states of the automata having the event in their event set. SC allows only one boolean mode of interaction, namely AND (shared event is enabled if it is enabled in the participating states of both automata), and is adequate to model the interaction of two subsystems within a physical system.

Interactions of the components of a DES can be achieved though synchronous composition (SC) [9, 12]. A DES requiring control is called a plant, and a supervisor is another DES that exercises control over the plant. SC is not adequate to model the control-interaction of a plant and a supervisor. The events occurring in the plant are partitioned into two disjoint sets - controllable events, which can be allowed to occur or prevented from occurring by supervisor agent, and uncontrollable events where no such control is possible. In the interaction of plant and supervisor the event set of the plant and the supervisor are the same, i.e., all events are shared, and so SC requires that all events occur synchronously, including the uncontrollable events. Since uncontrollable events occurring in the plant must never be disabled, a supervisor that interacts with a plant through SC must satisfy the structural requirement of control-compatibility, called \( \Sigma_u \)-enabling, which requires that in any state of the supervisor, a feasible uncontrollable event must be defined. In [3, 6], the supervisor is permitted to initiate certain events which the plant synchronously executes. Additionally in [2], events can originate both in the plant and the supervisor. In all cases the events are executed synchronously.

To relax the \( \Sigma_u \)-enabling requirement, a more general mechanism of interaction, the prioritized synchronous composition (PSC) [7], was proposed by Heymann. It extends the notion of SC by associating with systems event priority sets. Each of the interacting automata has its priority set of events, in whose occurrence the given automata must participate. PSC and its extension, masked PSC where systems interact with their environment via interface masks, have been applied to supervisory control in [1, 8, 10, 13, 14, 17, 18]. PSC can be viewed as an interaction which permits four modes of automata
interaction, that result from a four-way partitioning of the event set provided by the two priority sets. Letting $P$ denote a plant and $Q$ denote a supervisor, these four modes are “$p$”, “$q$”, “$\text{AND}$”, and “$\text{OR}$”.

While PSC is adequate to model interaction of plant and supervisor, additional modes of interactions are possible when local supervisors interact to determine their global control decisions. If we restrict the control decision fusion rule to be “enablement-based”, i.e., one where the global decision to enable an event is based on whether or not the event is locally enabled by at least one of the local supervisors, then it is known from boolean logic operations that these additional modes of interaction consist of “exclusive $P$”, “exclusive $Q$”, “exclusive $P$ or exclusive $Q$”, and “exclusive $P$ and exclusive $Q$”. Since these additional modes require exclusivity of participation we propose a generalization of PSC by introducing an exclusivity set besides the priority sets. The resulting composition is called prioritized composition with exclusion (PCX). The case of mutual exclusion when one requires that exactly one of the automata participate in the interaction, or the case of exclusive dominance of one automata over the other with regard to certain events, and also the case of certain events being blocked entirely, can now be addressed in the generalized interaction scheme of PCX.

However, even for automata interacting via PCX it is not possible for an event to exist in the composition if it is not defined in the participating state of both the interacting automata. If we allow events to exist in the composition under such a condition, then the complete set of sixteen boolean logic interactions become possible. In order to capture the eight additional interaction modes, we introduce a fourth set, called the nor-generative set. This set consists of events with the property that they are enabled in the composition without the enablement by either of the two automata (which is the logic of “nor” operation). Introduction of the nor-generative set besides the exclusivity set and priority sets results in the most general form of interaction, called prioritized composition with exclusion and generation (PCXG), wherein the restriction of global decision to enable an event to be enablement-based is relaxed as well. PCXG can thus be used to enable events in a global supervisor even when all the local ones do not permit the enablement of such an event. As is discussed further, such a composition mechanism offers a way for the reuse of existing supervisors to obtain a new one.

The rest of the paper is organized as follows: In section 2 preliminaries related to the theory and composition of DESs are discussed. This is followed by an alternative representation of prioritized synchronous composition in Section 3. In Section 4 prioritized composition with exclusion is defined and its use in supervisory control illustrated. Next, in Section 5 prioritized composition with exclusion and generation is defined and its applicability in decentralized decision fusion proposed. Section 6 discusses prioritized composition with ex-
clusion and generation of rules based models [5] using an example drawn from process control. Finally Section 7 provides conclusions and directions for future research.

2 NOTATION AND PRELIMINARIES

An overview of the automata based model [16] of discrete event systems, and supervisory control theory notation is presented here. Let \( \Sigma \) denote a finite set of events. A concatenation of events forms a string of events or a trace. A language is a collection of traces. Let \( \Sigma^* \) be the set of all finite strings (traces) of events of \( \Sigma \) including the empty string \( \epsilon \). A language is thus a subset of \( \Sigma^* \). Abstractly, a discrete event system can also be viewed as a state machine \( P = (X, \Sigma, \delta_P, x_0, X_m) \), where \( X \) is the set of states, \( \Sigma \) is the set of events, \( \delta_P : X \times \Sigma \to X \) is the partial state transition function, \( x_0 \in X \) is the initial state, and \( X_m \subseteq X \) is the set of marked or final states. The behavior of the discrete event system modeled by \( P \) is described by its generated language, \( L(P) := \{ s \in \Sigma^* | \delta_P(x_0, s) \text{ is defined} \} \), where by induction the transition function has been extended from events to traces \( \delta_P : X \times \Sigma^* \to X \). The generated language of \( P \) is the set of all traces that it can execute starting from its initial state.

The marked language of \( P \) contains those generated traces which terminate in a final state and signify task completion, \( L_m(P) := \{ s \in L(P) | \delta_P(x_0, s) \in X_m \} \).

Interaction of two DESs sharing events is modeled by their synchronous composition.

Definition 1 Synchronous Composition:

Given two deterministic automata \( P := (X, \Sigma_P, \delta_P, x_0, X_m) \) and \( Q := (Y, \Sigma_Q, \delta_Q, y_0, Y_m) \), the synchronous composition [9] of \( P \) and \( Q \) is denoted \( R = P \parallel Q := (Z, \Sigma, \delta, z_0, Z_m) \), where \( Z := X \times Y, \Sigma := \Sigma_P \cup \Sigma_Q, z_0 := (x_0, y_0), Z_m := X_m \times Y_m \), and the transition function \( \delta : Z \times \Sigma \to Z \) is defined as: \( \forall z = (x, y) \in Z, \sigma \in \Sigma, \)

\[
\delta((x, y), \sigma) := \begin{cases} 
(\delta_P(x, \sigma), \delta_Q(y, \sigma)) & \text{if } \delta_P(x, \sigma), \delta_Q(y, \sigma) \text{ defined}, \sigma \in \Sigma_P \cap \Sigma_Q \\
(\delta_P(x, \sigma), y) & \text{if } \delta_P(x, \sigma) \text{ defined}, \sigma \in \Sigma_P - \Sigma_Q \\
(x, \delta_Q(y, \sigma)) & \text{if } \delta_Q(y, \sigma) \text{ defined}, \sigma \in \Sigma_Q - \Sigma_P \\
\text{undefined} & \text{otherwise}
\end{cases}
\]

So when \( P \) and \( Q \) are composed, the common events occur synchronously, while the other events occur asynchronously. When \( \Sigma_P = \Sigma_Q = \Sigma \), then \( L(P \parallel Q) = L(P) \cap L(Q) \) and \( L_m(P \parallel Q) = L_m(P) \cap L_m(Q) \) since all events must occur synchronously and this composition is termed as strict synchronous composition.
Synchronous composition of plant automaton $P$ and supervisor automaton $Q$ may be used to achieve supervisory control, in which case the controlled plant is represented by the automaton $P||Q$. Since $Q$ must never prevent any feasible uncontrollable event from happening, $Q$ must satisfy the following: $L(P||Q)\Sigma_u\cap L(P) \subseteq L(P||Q)$, in which case $Q$ is said to be $\Sigma_u$-enabling \cite{12, 11}. In order to relax the structural requirement of $\Sigma_u$-enabling, SC was generalized to prioritized synchronous composition (PSC) by Heymann \cite{7}.

**Definition 2 Prioritized Synchronous Composition:**

Consider two automata, $P := (X, \Sigma, \delta_P, x_0, X_m)$, and $Q := (Y, \Sigma, \delta_Q, y_0, Y_m)$, with a common events set $\Sigma$. Let $A, B \subseteq \Sigma$ be the priority sets of $P$ and $Q$ respectively, then PSC of $P$ and $Q$ is denoted by $R = P_A||_B Q := (Z, \Sigma, \delta, z_0, Z_m)$, where $Z := X \times Y, z_0 := (x_0, y_0), Z_m := X_m \times Y_m$, and the transition function $\delta : Z \times \Sigma \rightarrow Z$ is defined as: $\forall z = (x, y) \in Z, \sigma \in \Sigma,$

$$\delta((x, y), \sigma) := \begin{cases} 
(\delta_P(x, \sigma), \delta_Q(y, \sigma)) & \text{if } \delta_P(x, \sigma), \delta_Q(y, \sigma) \text{ defined} \\
(\delta_P(x, \sigma), y) & \text{if } \delta_P(x, \sigma) \text{ defined, } \sigma \notin B \\
(x, \delta_Q(y, \sigma)) & \text{if } \delta_Q(y, \sigma) \text{ defined, } \sigma \notin A \\
\text{undefined} & \text{otherwise}
\end{cases}$$

3 AN ALTERNATIVE VIEW OF PSC

The event priority sets associated with two automata interacting via PSC can be viewed equivalently as a four-way partitioned set. When two automata, $P$ and $Q$, with events set $\Sigma$, interact under PSC via priority sets $A, B \subseteq \Sigma$ respectively, the event set is partitioned into four distinct subsets (see Figure 1):

1. The $P$-controlled event set, $A - B$, in which the events are executed at a state in the resultant automaton if and only if the event is permitted at the participating state of automaton $P$. This set of events we term as the $\Sigma_P$ partition.

2. The $Q$-controlled event set, $B - A$, obtained in a similar way to the P-controlled event set. This set of events we term as the $\Sigma_Q$ partition.

3. The $(P \land Q)$-controlled event set, $A \cap B$, in which the events are executed concurrently or not at all. This set of events we term as the $\Sigma_\land$ or the $\Sigma_{\text{AND}}$ partition.

4. The $(P \lor Q)$-controlled event set, $(A \cup B)^c$, in which the events are executed at a state in the resultant automaton if and only if at least one of the participating states of the automata, $P$ or $Q$, permit it. This set of events we term as the $\Sigma_\lor$ or $\Sigma_{\text{OR}}$ partition.
Table 1 summarizes the events in the four partitions, and their condition of enablement in the composed system. It follows from the entries in Table 1 that in the definition of PSC:

- An event can occur synchronously (i.e., the first clause can hold) for any event $\sigma \in \Sigma_p \cup \Sigma_q \cup \Sigma_{\land} \cup \Sigma_{\lor}$,
An event in $\Sigma_p \cup \Sigma_q$ can occur asynchronously in $P$ (i.e., the second clause can hold) for any event $\sigma \in \Sigma_p \cup \Sigma_q$.

An event in $\Sigma_q \cup \Sigma_v$ can occur asynchronously in $Q$ (i.e., the third clause can hold) for any event $\sigma \in \Sigma_q \cup \Sigma_v$.

So, the following definition of PSC is equivalent to the one used in Definition 2.

**Definition 3** Prioritized Synchronous Composition:

Consider two automata, $P := (X, \Sigma, \delta_P, x_0, X_m)$, and $Q := (Y, \Sigma, \delta_Q, y_0, Y_m)$, with a common events set $\Sigma$. Let $A, B \subseteq \Sigma$ be the priority sets of $P$ and $Q$ respectively, then the PSC of $P$ and $Q$ is denoted by $R = P_A \parallel B Q := (Z, \Sigma, \delta, z_0, Z_m)$, where $Z := X \times Y$, $z_0 := (x_0, y_0)$, $Z_m := X_m \times Y_m$, and the transition function $\delta : Z \times \Sigma \rightarrow Z$ is defined as: $\forall z = (x, y) \in Z$, $\sigma \in \Sigma$,

$$
\delta((x, y), \sigma) := \begin{cases} 
(\delta_P(x, \sigma), \delta_Q(y, \sigma)) & \text{if } \delta_P(x, \sigma), \delta_Q(y, \sigma) \text{ defined}, \\
(\delta_P(x, \sigma), y) & \text{if } \delta_P(x, \sigma) \text{ defined, } \delta_Q(y, \sigma) \text{ undefined}, \\
(x, \delta_Q(y, \sigma)) & \text{if } \delta_Q(y, \sigma) \text{ defined, } \delta_P(x, \sigma) \text{ undefined}, \\
\text{undefined} & \text{otherwise}
\end{cases}
$$

4 PRIORITIZED COMPOSITION WITH EXCLUSION (PCX)

We first present an example which illustrates certain modes of interaction that are not allowed by PSC.

**Example 1** Readers-writers database access:

Two reader/writer processes accessing a shared database system are modeled using automata $P$ and $Q$, shown in Figure 2. Each process has similar capabilities related to reading/writing of data from/to the database, and their event set is $\Sigma = \{r1, c1, w, r2, c2\}$. In both automata $r1$ represents the reading of data by the first process $P$, and $w$ represents the writing of data. Before the data is written, the write operation can be cleared using the operation $c1$. A similar set of operations $r2$, $c2$, and $w$ are also possible in the second process $Q$. To preserve the consistency of data during database transactions, exclusive locks, priority accesses, or lockouts of data, are frequently required. We consider next three particular scenarios involving interactions of $P$ and $Q$ processes. In Figure 2(b), (c), (d), the transitions that should be removed during
V. CHANDRA, Z. HUANG, W. QIU AND R. KUMAR

Fig. 2. Interacting automata

Each interaction scenario are represented by the dashed lines, whereas the solid lines depict the appropriate desired behavior of the composed system.

- As shown in Figure 2(b), suppose it is required that following exactly one read operation (r1 or r2), a write operation (w) be possible, but not after the data has been read in by both the processes. This scenario illustrates the need for an operator which captures a mutual exclusion type interaction.

- As shown in Figure 2(c), suppose it is required that the event w occur only when the first process (automaton P) attempts to execute it. Whenever the second process attempts to initiate the w operation, it should be blocked. Thus this effectively prevents the second process from executing the event w. This scenario illustrates the need for an operator which captures an ONLY P type interaction.

- As shown in Figure 2(d), suppose it is required that the event w be abandoned in the final automaton, despite attempts by either of the participating processes to execute it. This could be required when the write event has to be culled owing to considerations, such as housekeeping when data should not be altered, thereby leaving both processes with the abil-
ity to read from the database but not to write to it at all. This scenario illustrates the need for an operator which captures a \( \emptyset \) type interaction.

Since PSC allows for two priority sets, \( A \) and \( B \), exactly four partitions of the event set of the two automata are possible. However, these form only a part of the complete set of interactions possible when two automata interact. From boolean logic operations, we know that other interactions, such as XOR, \( \text{ONLY } P \equiv xP \), \( \text{ONLY } Q \equiv xQ \), and \( \emptyset \equiv \land \) are also possible. These latter interactions occur when exclusively one automaton participates in the interaction. So, to accommodate them, we introduce a third set, called the *exclusivity set*, denoted \( C \), in addition to the priority sets \( A \) and \( B \). The event set \( C \) consists of events whose occurrence in the composition require the participation of exclusively one automaton. Introduction of the set \( C \), creates four new partitions of the event set \( \Sigma \) (see Figure 3). A brief description of each partition follows.

5. \( (P \times \land Q)\) controlled event set, \( ((A - B) \cap C) \), which signifies that the event exists in the resultant automaton if and only if it exists in only the participating state of the \( P \) automaton.

6. \( (P \times Q)\) controlled event set, \( ((B - A) \cap C) \), which signifies that the event exists in the resultant automaton if and only if it exists in only the participating states of \( Q \) automaton.

7. \( (P \times \emptyset \land Q)\) controlled event set, \( ((A \cap B) \cap C) \) which signifies that the event exists in the resultant automaton if and only if it exists in the participating state of only the \( P \) automaton, and also in the participating state of only the \( Q \) automaton. This represents an impossibility, and is denoted as \( \land \text{AND} \) or \( \emptyset \). Thus an event belonging to the \( \emptyset \) set does not exist in the resultant automaton regardless of whether it exists in the participating states of the interacting automata.

8. \( (P \times \lor Q)\) controlled event set, \( ((A \cup B)^c \cap C) \), which signifies that the event exists in the resultant automaton if and only if it is present in exactly one of the participating states of the two interacting automata.

Figure 3 summarizes all these eight types of boolean interactions, and in Table 2 the conditions for enablement of events in four new partitions (5 through 8) are defined in addition to the four previously defined ones in Table 1. For example, events in the set \( \Sigma_{\times P} \) belong to the priority set of \( P \), and not to the priority set of \( Q \), and to that of the exclusivity set. So, such events are enabled if and only if \( P \) can participate exclusively (i.e., \( Q \) cannot participate). These eight types of interactions possible between two automata can be modeled using three sets, \( A \), \( B \) and \( C \), as shown in Figure 3. Conversely, given these eight partitions, the three sets \( A \), \( B \) and \( C \) can be derived as follows:
1. \( A = \Sigma_p \cup \Sigma_\wedge \cup \Sigma_\emptyset \cup \Sigma_{xp}. \)
2. \( B = \Sigma_q \cup \Sigma_\wedge \cup \Sigma_\emptyset \cup \Sigma_{xq}. \)
3. \( C = \Sigma_\oplus \cup \Sigma_{xp} \cup \Sigma_{xq} \cup \Sigma_\emptyset. \)

<table>
<thead>
<tr>
<th>No.</th>
<th>Partition</th>
<th>PCX Partitions</th>
<th>Enabled if and only if</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \Sigma_p )</td>
<td>( (A - B) - C )</td>
<td>p-enabled</td>
</tr>
<tr>
<td>2</td>
<td>( \Sigma_q )</td>
<td>( (B - A) - C )</td>
<td>q-enabled</td>
</tr>
<tr>
<td>3</td>
<td>( \Sigma_\wedge )</td>
<td>( (A \cap B) - C )</td>
<td>((p \wedge q))-enabled</td>
</tr>
<tr>
<td>4</td>
<td>( \Sigma_\lor )</td>
<td>( (A \cup B)^c - C )</td>
<td>((p \lor q))-enabled</td>
</tr>
<tr>
<td>5</td>
<td>( \Sigma_{xp} )</td>
<td>( (A - B) \cap C )</td>
<td>Exclusive P-enabled</td>
</tr>
<tr>
<td>6</td>
<td>( \Sigma_{xq} )</td>
<td>( (B - A) \cap C )</td>
<td>Exclusive Q-enabled</td>
</tr>
<tr>
<td>7</td>
<td>( \Sigma_\emptyset )</td>
<td>( (A \cap B) \cap C )</td>
<td>FALSE</td>
</tr>
<tr>
<td>8</td>
<td>( \Sigma_\oplus )</td>
<td>( (A \cup B)^c \cap C )</td>
<td>((p \oplus q))-enabled</td>
</tr>
</tbody>
</table>

Fig. 3. Partitions under PCX

It follows from the entries in Table 2 that:

- **An event occurs synchronously in the two automata if it does not belong to the exclusivity set, i.e., it belongs to \( \Sigma_p \cup \Sigma_q \cup \Sigma_\wedge \cup \Sigma_\lor. \)**
- **An event occurs asynchronously in the \( P \) automaton if it belongs to \( \Sigma_{xp} \cup \Sigma_\emptyset \cup \Sigma_\oplus \cup \Sigma_\lor. \)**
- **An event occurs asynchronously in the \( Q \) automaton if it belongs to \( \Sigma_{xq} \cup \Sigma_\emptyset \cup \Sigma_\oplus \cup \Sigma_\lor. \)**
- **An event does not occur if it belongs to \( \Sigma_\emptyset. \)**
<table>
<thead>
<tr>
<th>No.</th>
<th>Partition</th>
<th>Enablement conditions</th>
</tr>
</thead>
</table>
| 5   | $\Sigma_{xp}$ | $\sigma \in \Sigma \mid (\delta_P(x, \sigma) \text{ defined}) \iff [\delta((x, y), \sigma) \text{ defined}]$  
   |   | Equivalently:  
   |   | $\{ \sigma \in \Sigma \mid (\delta_P(x, \sigma) \text{ defined}) \land (\delta_Q(y, \sigma) \text{ undefined})\}$  
   |   | $\iff [\delta((x, y), \sigma) \text{ defined}]$  |
| 6   | $\Sigma_{xq}$ | $\sigma \in \Sigma \mid (\delta_P(x, \sigma) \text{ defined}) \lor (\delta_Q(y, \sigma) \text{ defined})$  
   |   | Equivalently:  
   |   | $\{ \sigma \in \Sigma \mid (\delta_P(x, \sigma) \text{ undefined}) \land (\delta_Q(y, \sigma) \text{ defined})\}$  
   |   | $\iff [\delta((x, y), \sigma) \text{ defined}]$  |
| 7   | $\Sigma_{\Theta}$  
   |   | or  
   |   | $\Sigma_{xand}$ | $\sigma \in \Sigma \mid (\delta_P(x, \sigma) \text{ defined}) \land (\delta_Q(y, \sigma) \text{ defined})$  
   |   | Equivalently:  
   |   | $\{ \sigma \in \Sigma \mid (\delta_P(x, \sigma) \text{ defined/undefined}) \lor (\delta_Q(y, \sigma) \text{ defined/undefined})\}$  
   |   | $\iff [\delta((x, y), \sigma) \text{ undefined}]$  |
| 8   | $\Sigma_{xor}$  
   |   | or  
   |   | $\Sigma_{@}$ | $\sigma \in \Sigma \mid (\delta_P(x, \sigma) \text{ defined}) \lor (\delta_Q(y, \sigma) \text{ defined})$  
   |   | Equivalently:  
   |   | $\{ \sigma \in \Sigma \mid ((\delta_P(x, \sigma) \text{ defined}) \land (\delta_Q(y, \sigma) \text{ undefined}) \lor ((\delta_P(x, \sigma) \text{ undefined}) \land (\delta_Q(y, \sigma) \text{ defined}))\}$  
   |   | $\iff [\delta((x, y), \sigma) \text{ defined}]$  |

Table 2. Partitions defined under PCX

So, we have the following formal definition of PCX.

**Definition 4** *Prioritized Composition with Exclusion:*

Given two systems $P := (X, \Sigma, \delta_P, x_0, X_m)$ and $Q := (Y, \Sigma, \delta_Q, y_0, Y_m)$, with their respective priority sets as $A$ and $B$, along with an exclusivity set $C$, the prioritized composition with exclusion of $P$ and $Q$ is denoted by $R = P \wedge_C Q := (Z, \Sigma, \delta, z_0, Z_m)$, where $Z := X \times Y, z_0 := (x_0, y_0), Z_m := X_m \times Y_m$, and the
transition function $\delta : Z \times \Sigma \rightarrow Z$ is defined as: $\forall z = (x, y) \in Z$, $\sigma \in \Sigma$, 

$$
\delta((x, y), \sigma) := \begin{cases} 
(\delta_P(x, \sigma), \delta_Q(y, \sigma)) & \text{if } \delta_P(x, \sigma), \delta_Q(y, \sigma) \text{ defined,} \\
(\delta_P(x, \sigma), y) & \text{if } \delta_P(x, \sigma) \text{ defined, } \delta_Q(y, \sigma) \text{ undefined,} \\
(x, \delta_Q(y, \sigma)) & \text{if } \delta_Q(y, \sigma) \text{ defined, } \delta_P(x, \sigma) \text{ undefined,} \\
\text{undefined} & \text{otherwise}
\end{cases}
$$

where the sets $\Sigma_p, \Sigma_q, \Sigma_{xp}, \Sigma_{xp}, \Sigma_{xp}, \Sigma_{xp}, \Sigma_{xp}$ and $\Sigma_\emptyset$ in the definition of PCX form a partition of the event set $\Sigma$ into eight distinct subsets given above.

4.1 Properties of PCX

**Property 1** When $\Sigma_\emptyset = \Sigma$ (i.e., the sets $\Sigma_{\emptyset} = \Sigma_{xp} = \Sigma_{xp} = \Sigma_\emptyset = \Sigma_p = \Sigma_q = \Sigma = \Sigma_\emptyset = \emptyset$) then PCX reduces to synchronous composition (SC).

If we set the elements in the partition $\Sigma_\emptyset = \Sigma$ in the transition function of PCX given in Definition 4, then it is clear that the definition of PCX reduces to that of SC given in Definition 1.

**Property 2** When the sets $\Sigma_{\emptyset} = \Sigma_{xp} = \Sigma_{xp} = \Sigma_\emptyset = \emptyset$, PCX reduces to PSC.

If we set the elements in the partitions $\Sigma_{\emptyset} = \Sigma_{xp} = \Sigma_{xp} = \Sigma_\emptyset = \emptyset$ in the transition function of PCX given in Definition 4, then it is clear that the definition of PCX reduces to that of PSC given in Definition 3.

**Property 3** PCX is commutative.

This follows from the definition of PCX. Since the sets involved in a PCX based interaction are decided prior to the composition, the order in which the two automata are composed is immaterial, i.e., $P_{A \uparrow B} Q = Q_{B \uparrow A} P$.

**Property 4** Self-loop augmentation on $\Sigma_h$ in PCX

Let $P$ be an automaton with event set $\Sigma$, and let $\Sigma_h \subseteq \Sigma$. Let $H$ be a deterministic automaton having one state and with self-loops labeled by every event in $\Sigma_h$. The augmentation of $P$ by $\Sigma_h$, denoted by $P^{\Sigma_h}$, is defined to be the automaton $P^{\Sigma_h} := P_{A \uparrow B} H$, with $\Sigma = \Sigma_h$, and the remaining 7 sets of PCX as empty sets.

The state space of $P^{\Sigma_h}$ will be identical to the state space of $P$, and $P^{\Sigma_h}$ is obtained by adding self-loops at each state on all events in $\Sigma_h$ except those already defined at that state.
4.2 Comments on PSC vs. PCX

Some comments on the interaction of automata using PSC and PCX are given next:

- Under PSC each event in the system is forced to occur synchronously whenever possible, i.e., whenever the event is feasible in the participating states of all automata. This is more permissive than may be required for certain interactions. When automata representing design specifications of a DES interact amongst themselves, they might place restrictions on the synchronous or otherwise occurrence of certain events. Such requirement can be accommodated via a PCX interactions which permits events to occur asynchronously, even when they are defined at the participating states of all automata.

- In PSC the priority sets of the interacting automata are combined using set union to obtain the priority set of the composed automaton. Under certain conditions the priority sets of the interacting automata may depend on the interaction itself, and need not equal the union of the priority sets of the interacting automata. This is permitted under PCX, where depending on the type of interaction needed, the priority sets and exclusivity set are chosen independent of any preceding priority sets and exclusivity set used for the interaction of the underlying automata.

4.3 Using PCX for Supervisory Composition

PCX can be used to represent composition of local supervisors to obtain a global one. The purpose of the composition is to fuse the control decision of local supervisors to obtain the global control decision. For example, if an event belongs to the set XOR, then it will be globally enabled if and only if it is locally enabled by exactly one of the supervisors. PCX allows for eight such control decision fusion rules, and exhausts all the possible eight binary modes of composition that are enablement-based, i.e., in which, an event is enabled globally only if it is enabled locally by at least one local supervisor. Note that PCX does not allow for event “creation”, wherein an event is enabled globally even if it is disabled locally by each local supervisor. Such a restriction may be a natural one for certain applications, however, we do relax this later when we extend PCX to define prioritized composition with exclusion and generation (PCXG). PCX thus serves as a powerful tool for combining local supervisors in order to build up a combined global supervisor for a given DES, in ways not possible using any existing composition method. Once a global supervisor has been obtained from the local supervisors, it can be used in conjunction with the plant, using PSC for supervisory control purposes.
Example 2 \textit{PCX} in supervisor computation:

In Figure 4(a), (b), and (c), a plant and two of its local supervisors are shown. The events occurring in the plant are $\Sigma = \{a, b, c, d, e\}$. The two supervisors
$S1 = P$, and $S2 = Q$ are combined first using a specific PCX operation, such that the event $a$ occurs synchronously, $b$ when only the local supervisor $S1$ permits it, $c$ when exactly one of the local supervisors permits it, $d$ when the local supervisor $S1$ permits it, and finally $e$ is always blocked. This sort of interaction can be achieved by placing the events in the following PCX partitions: $\Sigma^\Lambda = \{a\}$, $\Sigma^\Sigma_P = \{b\}$, $\Sigma^\Sigma_Q = \{c\}$, $\Sigma^\Sigma_r = \{d\}$, $\Sigma^\Sigma_o = \{e\}$, $\Sigma^\Sigma_d = \Sigma^\Sigma_o = \Sigma^\Sigma = \{\}$. The resultant of this composition, the combined supervisor automaton, is shown in Figure 4(d). Next, this combined supervisor automaton (Figure 4(d)) interacts with the plant automaton (Figure 4(a)) for the purpose of supervisory control. For this PCX interaction the plant automaton is designated as $P$ and the combined supervisor automaton as $Q$. Now, when the uncontrollable events are placed in the $\Sigma^\Sigma_r$ partition and the controllable events in the $\Sigma^\Lambda$ partition, the plant can execute the uncontrollable event transitions without any restriction by the supervisor. Also, all the controllable events are synchronized between the plant and the supervisor. In Figure 4(a), (d) the controllable events in the plant are $\{a, c, d\}$ and the uncontrollable events are $\{b, e\}$. By choosing $\Sigma^\Lambda = \{a, c, d\}$ and $\Sigma^\Sigma_r = \{b, e\}$, the supervised plant shown in Figure 4(e) is obtained.

5 PRIORITIZED COMPOSITION
WITH EXCLUSION AND GENERATION

In addition to the eight modes of interaction specified using PCX, it is possible for two local supervisors to interact in eight additional modes, in such a way that even when an event is disabled in both the participating states of the local supervisors, yet the event is enabled in the composition. Such an event appears as a self-loop in the composition, i.e., no state change occurs when this event occurs in the composition. This sort of event generation provides additional flexibility in the reuse of existing supervisors to obtain a new combined one. The new resulting composition mechanism is what we call, prioritized composition with exclusion and generation (PCXG).

Example 3 Readers-writers with data backup:

In the readers-writers example suppose it is required that regardless the state of the overall system, the data backup operations (say an event $b$) be permitted by a global supervisor, even when none of the local supervisors permit it. This scenario is illustrated in the desired composition of $P$ and $Q$ of Figure 5(a), (b). The PCXG of $P$ and $Q$ may be used resulting in the addition of the event $b$ as a self-loop on all the states of the combined system wherever the event $b$ is disabled at all the participating states.
The additional eight modes of interaction are possible by introducing a new set \( D \), called the \textit{nor-generative set}. The events which belong to set \( D \) have the property that they are enabled in the composition when the interacting automata disable them (which is the logic of “nor” operation). The PCXG interaction which is defined in terms of the priority sets \( A, B \), the exclusivity set \( C \), and the nor-generative set \( D \) of automata \( P \) and \( Q \), partitions the event set \( \Sigma \) into sixteen disjoint subsets: In the presence of nonempty \( D \), each of the eight event set partitions under PCX (see Figure 3), say \( \Sigma_{OP} \), is further partitioned into: \( \Sigma_{OP} - D \) and \( \Sigma_{OP} \cap D \). In PCXG, the semantics of \( \Sigma_{OP} - D \) remains the same as that of \( \Sigma_{OP} \) in PCX, whereas an event in \( \Sigma_{OP} \cap D \) is enabled under PCXG when either it is enabled under PCX, or it is disabled by both \( P \) and \( Q \). Thus the semantics of \( \Sigma_{OP} \cap D \) under PCXG is that of \( \Sigma_{OP} \lor \text{NOR} \), which we denote as \( \Sigma_{OP+} \), and refer to it as the \textit{nor-disjunct} of \( \Sigma_{OP} \). For example, under PCX the \( \text{xp} \) partition, which is defined to be \( [(A - B) \cap C] \) using the sets \( A, B \) and \( C \), owing to the introduction of a nonempty \( D \), splits into \( [(A - B) \cap C] - D \) and \( [(A - B) \cap C] \cap D \), denoted \( \Sigma_{XP} \) and \( \Sigma_{XP+} \) respectively.

A brief description of such newly formed eight partitions follows:

9. The \((P \ + \ Q)-controlled \ \textit{event set}, \ [(A - B) - C] \cap D \), in which the event exists in the composition if and only if either it exists in the participating state of \( P \) automaton, or if it does not exist in both participating states of \( P \) and \( Q \) automata. This partition we designate as \( \Sigma_{P+} \).
10. The \((P Q^+)\)-controlled event set, \([(B - A) - C] \cap D\), in which the event exists in the composition if and only if either it exists in the participating state of \(Q\) automaton, or if it does not exist in both participating states of \(P\) and \(Q\) automata. This partition we designate as \(\Sigma_{q^+}\).

11. The \((P \land^+ Q)\)-controlled event set, \([(A \cap B) - C] \cap D\), in which the event exists in the composition if and only if either it exists in the both or neither of the participating states of \(P\) and \(Q\) automata. This partition we designate as \(\Sigma_{\land^+}\).

12. The \((P \lor^+ Q)\)-controlled event set, \([(A \cup B)^c - C] \cap D\), in which the event always always exists in the composition regardless of whether it exists in the participating states of the \(P\) and \(Q\) automata. This partition we designate as \(\Sigma_{\lor^+}\).

13. The \((P \times^+ Q)\)-controlled event set, \([(A - B) \cap C] \cap D\), in which the event exists in the composition if and only if it exists in only the participating state of \(P\) automaton, or if does not exist in both participating states of \(P\) and \(Q\) automata. This partition we designate as \(\Sigma_{\times^+}\).

14. The \((P \times Q^+)\)-controlled event set, \([(B - A) \cap C] \cap D\), in which the event exists in the composition if and only if either it exists in exactly one of the participating states of \(P\) and \(Q\) automata, or if it does not exist in both participating states of the \(P\) and \(Q\) automata. This partition we designate as \(\Sigma_{\times^+}\).

15. The \((P \Theta^+ Q)\)-controlled event set, \([(A \cap B) \cap C] \cap D\), in which the event exists in the composition if and only if it does not exist in both participating states of \(P\) and \(Q\) automata. This partition we designate as \(\Sigma_{\Theta^+}\).

16. The \((P \Theta^+ Q)\)-controlled event set, \([(A \cup B)^c - C] \cap D\), in which the event exists in the composition if and only if it exists in exactly one of the participating states of \(P\) and \(Q\) automata, or if it does not exist in both participating states of the \(P\) and \(Q\) automata. This partition we designate as \(\Sigma_{\Theta^+}\).

Figure 6 summarizes all the sixteen partitions of the event set, only the first four of which are present under PSC, and the first eight under PCX.

Conversely, given these sixteen partitions, the four sets \(A\), \(B\), \(C\), and \(D\) can be defined as follows:

1. \(A = \Sigma_{p} \cup \Sigma_{\land} \cup \Sigma_{\lor} \cup \Sigma_{\times} \cup \Sigma_{p^+} \cup \Sigma_{\land^+} \cup \Sigma_{\lor^+} \cup \Sigma_{\times^+}\).
2. \(B = \Sigma_{q} \cup \Sigma_{\land} \cup \Sigma_{\lor} \cup \Sigma_{\times} \cup \Sigma_{q^+} \cup \Sigma_{\land^+} \cup \Sigma_{\lor^+} \cup \Sigma_{\times^+}\).
3. \(C = \Sigma_{\Theta} \cup \Sigma_{\times} \cup \Sigma_{\times} \cup \Sigma_{\Theta^+} \cup \Sigma_{\times^+} \cup \Sigma_{\Theta^+} \cup \Sigma_{\times^+} \cup \Sigma_{\Theta^+}\).
### Fig. 6. Partitions under PCXG

<table>
<thead>
<tr>
<th>No.</th>
<th>Partition</th>
<th>PCXG Partitions</th>
<th>Enabled if and only if</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\Sigma_p$</td>
<td>$(A - B) - C - D$</td>
<td>p-enabled</td>
</tr>
<tr>
<td>2</td>
<td>$\Sigma_q$</td>
<td>$(B - A) - C - D$</td>
<td>Q-enabled</td>
</tr>
<tr>
<td>3</td>
<td>$\Sigma_{\wedge}$</td>
<td>$(A \cap B) - C - D$</td>
<td>(P &amp; Q)-enabled</td>
</tr>
<tr>
<td>4</td>
<td>$\Sigma_{\lor}$</td>
<td>$(A \cup B)^c - C - D$</td>
<td>(P \lor Q)-enabled</td>
</tr>
<tr>
<td>5</td>
<td>$\Sigma_{\oplus}$</td>
<td>$(A - B) \cap C - D$</td>
<td>Exclusively p-enabled</td>
</tr>
<tr>
<td>6</td>
<td>$\Sigma_{\ominus}$</td>
<td>$(B - A) \cap C - D$</td>
<td>Exclusively Q-enabled</td>
</tr>
<tr>
<td>7</td>
<td>$\Sigma_{\oplus^*}$</td>
<td>$(A \cap B) \cap C - D$</td>
<td>FALSE</td>
</tr>
<tr>
<td>8</td>
<td>$\Sigma_{\ominus^*}$</td>
<td>$(A \cup B)^c \cap C - D$</td>
<td>(P \oplus Q)-enabled</td>
</tr>
<tr>
<td>9</td>
<td>$\Sigma_{\oplus^+}$</td>
<td>$(A - B) - C \cap D$</td>
<td>Case 1 $\lor(P \land Q)$-disabled</td>
</tr>
<tr>
<td>10</td>
<td>$\Sigma_{\ominus^+}$</td>
<td>$(B - A) - C \cap D$</td>
<td>Case 2 $\lor(P \land Q)$-disabled</td>
</tr>
<tr>
<td>11</td>
<td>$\Sigma_{\wedge^+}$</td>
<td>$(A \cap B) - C \cap D$</td>
<td>Case 3 $\lor(P \land Q)$-disabled</td>
</tr>
<tr>
<td>12</td>
<td>$\Sigma_{\lor^+}$</td>
<td>$(A \cup B)^c - C \cap D$</td>
<td>Case 4 $\lor(P \land Q)$-disabled</td>
</tr>
<tr>
<td>13</td>
<td>$\Sigma_{\oplus^+}$</td>
<td>$(A - B) \cap C \cap D$</td>
<td>Case 5 $\lor(P \land Q)$-disabled</td>
</tr>
<tr>
<td>14</td>
<td>$\Sigma_{\ominus^+}$</td>
<td>$(B - A) \cap C \cap D$</td>
<td>Case 6 $\lor(P \land Q)$-disabled</td>
</tr>
<tr>
<td>15</td>
<td>$\Sigma_{\oplus^*}$</td>
<td>$(A \cap B) \cap C \cap D$</td>
<td>Case 7 $\lor(P \land Q)$-disabled</td>
</tr>
<tr>
<td>16</td>
<td>$\Sigma_{\ominus^*}$</td>
<td>$(A \cup B)^c \cap C \cap D$</td>
<td>Case 8 $\lor(P \land Q)$-disabled</td>
</tr>
</tbody>
</table>
4. $D = \Sigma_{p^+} \cup \Sigma_{q^+} \cup \Sigma_{A^+} \cup \Sigma_{V^+} \cup \Sigma_{KP^+} \cup \Sigma_{KQ^+} \cup \Sigma_{Q^+} \cup \Sigma_{\oplus^+}$.

The rules for enablement of events in the eight new partitions is given in Tables 3, and 4. Nor-disjuncts of the four PSC partitions are given in Table 3, whereas nor-disjuncts of the remaining four PCX partitions are given in Table 4.

It follows from the entries in Tables 3 and 4 that:

<table>
<thead>
<tr>
<th>No.</th>
<th>Partition</th>
<th>Enablement conditions</th>
</tr>
</thead>
</table>
| 9   | $\Sigma_{p^+}$ | \{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined}) \land (\delta_p(y, \sigma) \text{ defined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\} |

Equivalently:
\{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined}) \land (\delta_p(y, \sigma) \text{ defined/undefined})
\right]
\lor \left[
(f(x, \sigma) \text{ undefined}) \land (\delta_p(y, \sigma) \text{ undefined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\}

| 10  | $\Sigma_{q^+}$ | \{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined}) \land (\delta_q(y, \sigma) \text{ defined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\} |

Equivalently:
\{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined/undefined}) \land (\delta_q(y, \sigma) \text{ defined})
\right]
\lor \left[
(f(x, \sigma) \text{ undefined}) \land (\delta_q(y, \sigma) \text{ undefined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\}

| 11  | $\Sigma_{A^+}$ or $\Sigma_{AND^+}$ | \{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined}) \land (\delta_q(y, \sigma) \text{ defined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\} |

Equivalently:
\{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined}) \land (\delta_q(y, \sigma) \text{ defined/undefined})
\right]
\lor \left[
(f(x, \sigma) \text{ undefined}) \land (\delta_q(y, \sigma) \text{ undefined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\}

| 12  | $\Sigma_{V^+}$ or $\Sigma_{OR^+}$ | \{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined}) \lor (\delta_q(y, \sigma) \text{ defined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\} |

Equivalently:
\{
\sigma \in \Sigma \mid \left[
(f(x, \sigma) \text{ defined/undefined})
\lor (\delta_q(y, \sigma) \text{ defined/undefined})
\right]
\Rightarrow \delta(x, y, \sigma) \text{ defined}
\}

Table 3. Partitions 9-12 defined under PCXG

- An event occurs synchronously in the two automata if it belongs to $\Sigma_{p^+} \cup$
<table>
<thead>
<tr>
<th>No.</th>
<th>Partition</th>
<th>Enablement conditions</th>
</tr>
</thead>
</table>
| 13  | $\Sigma_{xp^+}$ | \{ $\sigma \in \Sigma$ | [(\(\delta_P(x, \sigma)\) defined) \(x_{p^+}\) (\(\delta_Q(y, \sigma)\) defined)] \} \iff [\(\delta((x, y), \sigma)\) defined] \\ *Equivalently:* \{ $\sigma \in \Sigma$ | \{[(\(\delta_P(x, \sigma)\) defined) \& (\(\delta_Q(y, \sigma)\) undefined)] \} \vee [(\(\delta_P(x, \sigma)\) undefined) \& (\(\delta_Q(y, \sigma)\) undefined)] \} \} \iff [\(\delta((x, y), \sigma)\) defined] \\ or, \{ \}$
| 14  | $\Sigma_{xq^+}$ | \{ $\sigma \in \Sigma$ | [(\(\delta_P(x, \sigma)\) defined) \(x_{q^+}\) (\(\delta_Q(y, \sigma)\) defined)] \} \iff [\(\delta((x, y), \sigma)\) defined] \\ *Equivalently:* \{ $\sigma \in \Sigma$ | \{[(\(\delta_P(x, \sigma)\) undefined) \& (\(\delta_Q(y, \sigma)\) defined)] \} \vee [(\(\delta_P(x, \sigma)\) undefined) \& (\(\delta_Q(y, \sigma)\) undefined)] \} \} \iff [\(\delta((x, y), \sigma)\) defined] \\ or, \{ \}$
| 15  | $\Sigma_{\oplus^+}$ | \{ $\sigma \in \Sigma$ | [(\(\delta_P(x, \sigma)\) defined) \(\ominus^+\) (\(\delta_Q(y, \sigma)\) defined)] \} \iff [\(\delta((x, y), \sigma)\) defined] \\ *Equivalently:* \{ $\sigma \in \Sigma$ | \{[(\(\delta_P(x, \sigma)\) undefined) \& (\(\delta_Q(y, \sigma)\) defined)] \} \vee [(\(\delta_P(x, \sigma)\) defined) \& (\(\delta_Q(y, \sigma)\) undefined)] \} \} \iff [\(\delta((x, y), \sigma)\) defined] \\ or, \{ \}$
| 16  | $\Sigma_{\boxplus^+}$ or $\Sigma_{\text{XOR}^+}$ | \{ $\sigma \in \Sigma$ | [(\(\delta_P(x, \sigma)\) defined) \(\oplus^+\) (\(\delta_Q(y, \sigma)\) defined)] \} \iff [\(\delta((x, y), \sigma)\) defined] \\ *Equivalently:* \{ $\sigma \in \Sigma$ | \{[(\(\delta_P(x, \sigma)\) undefined) \& (\(\delta_Q(y, \sigma)\) defined)] \} \vee [(\(\delta_P(x, \sigma)\) defined) \& (\(\delta_Q(y, \sigma)\) undefined)] \} \} \iff [\(\delta((x, y), \sigma)\) defined] \\ or, \{ \}$

Table 4. Partitions 13-16 defined under PCXG

\[\Sigma_{p^+} \cup \Sigma_{q^+} \cup \Sigma_{\lambda^+} \cup \Sigma_{\lambda^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+}.\]

- An event occurs asynchronously in the $P$ automata if it belongs to $\Sigma_{p^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\lambda^+} \cup \Sigma_{\lambda^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} .$

- An event occurs asynchronously in the $Q$ automata if it belongs to $\Sigma_{q^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} \cup \Sigma_{\nu^+} .$
Given two systems \( P := (X, \Sigma, \delta_P, x_0, X_m) \) and \( Q := (Y, \Sigma, \delta_Q, y_0, Y_m) \), with their respective priority sets as \( A \) and \( B \), along with an exclusivity set \( C \), and a nor-generative set \( D \), the prioritized composition with exclusion and generation of \( P \) and \( Q \) is denoted by \( R = P \uplus_{A, B}^{C, D} Q := (Z, \Sigma, \delta, z_0, Z_m) \), where \( Z := X \times Y \), \( z_0 := (x_0, y_0) \), \( Z_m := X_m \times Y_m \), and the transition function \( \delta : Z \times \Sigma \to Z \) is defined as: \( \forall z = (x, y) \in Z \), \( \sigma \in \Sigma \),

\[
\delta((x, y), \sigma) := \begin{cases} 
(\delta_p(x, \sigma), \delta_q(y, \sigma)) & \text{if } \delta_p(x, \sigma), \delta_q(y, \sigma) \text{ defined,} \\
\sigma \in \Sigma_\Sigma \cup \Sigma_{q^+} \cup \Sigma_q \cup \Sigma_{q^+} \cup \Sigma_\Lambda \cup \Sigma_\Lambda^+ \cup \Sigma_v \cup \Sigma_v^+ & \text{if } \delta_p(x, \sigma) \text{ defined, } \delta_q(y, \sigma) \text{ undefined,} \\
(\delta_p(x, \sigma), y) & \text{if } \delta_p(x, \sigma) \text{ undefined, } \delta_q(y, \sigma) \text{ defined,} \\
\sigma \in \Sigma_q \cup \Sigma_{q^+} \cup \Sigma_v \cup \Sigma_v^+ \cup \Sigma_{xq} \cup \Sigma_{xq^+} \cup \Sigma_q \cup \Sigma_{q^+} \cup \Sigma_v \cup \Sigma_v^+ & \text{if } \delta_p(x, \sigma) \text{ undefined, } \delta_q(y, \sigma) \text{ undefined,} \\
(x, y) & \text{otherwise}
\end{cases}
\]

where the sets \( \Sigma_p, \Sigma_{p^+}, \Sigma_q, \Sigma_{q^+}, \Sigma_\Lambda, \Sigma_{\Lambda^+}, \Sigma_v, \Sigma_{v^+}, \Sigma_{xq}, \Sigma_{xq^+}, \Sigma_\theta, \Sigma_{\theta^+}, \Sigma_{\theta^+}, \Sigma_{q^+}, \Sigma_{p^+}, \Sigma_{v^+}, \Sigma_{xq^+}, \Sigma_{xq^+}, \Sigma_{p^+}, \Sigma_{v^+}, \Sigma_{xq^+}, \Sigma_{xq^+}, \Sigma_{p^+} \), and \( \Sigma_{v^+} \), in the definition of PCXG, partition the event set \( \Sigma \) into sixteen distinct subsets given earlier in this section.

### 5.1 Properties of PCXG

**Property 5** When the partition \( \Sigma_\Lambda = \Sigma \), PCXG reduces to synchronous composition (SC).

If we set the elements in the partition \( \Sigma_\Lambda = \Sigma \) in the transition function of PCXG given in Definition 5, the transition function of PCXG reduces to that of SC given in Definition 1.

**Property 6** When the sets \( \Sigma_{p^+} = \Sigma_{q^+} = \Sigma_{\Lambda^+} = \Sigma_{v^+} = \Sigma_{xq^+} = \Sigma_\theta^+ = \Sigma_{\theta^+} = \{\} \), PCXG reduces to PCX.

If we set the elements in the partitions \( \Sigma_{p^+} = \Sigma_{q^+} = \Sigma_{\Lambda^+} = \Sigma_{v^+} = \Sigma_{xq^+} = \Sigma_\theta^+ = \Sigma_{\theta^+} = \{\} \) in the transition function of PCXG given in Definition 5, the transition function of PCXG reduces to that of PCX given in Definition 4.
Property 7 When the sets $\Sigma = \Sigma_x = \Sigma_q = \Sigma_\emptyset = \Sigma_{xq} = \Sigma_{xq^+} = \Sigma_{xq^+} + \Sigma_{q^+} = \Sigma_{q^+} + \Sigma_{p^+} = \Sigma_{p^+} + \Sigma_{\emptyset^+} = \Sigma_{\emptyset^+} = \{\}$, PCXG reduces to PSC.

If we set the elements in the partitions $\Sigma = \Sigma_x = \Sigma_q = \Sigma_\emptyset = \Sigma_{xq} = \Sigma_{xq^+} = \Sigma_{xq^+} + \Sigma_{q^+} = \Sigma_{q^+} + \Sigma_{p^+} = \Sigma_{p^+} + \Sigma_{\emptyset^+} = \Sigma_{\emptyset^+} = \{\}$ in the transition function of PCXG given in Definition 5, the transition function of PCXG reduces to that of PSC given in Definition 3. 

Property 8 PCXG is commutative.

This follows from the definition of PCXG. Since the sets involved in a PCXG based interaction are decided prior to the composition, the order in which the two automata are composed is immaterial, i.e., $P_{\mathcal{A} \mathcal{B}} Q = Q_{\mathcal{B} \mathcal{A}} P$. 

Property 9 Self-loop augmentation on $\Sigma_h$ in PCXG.

Let $P$ be an automaton with event set $\Sigma$, and let $\Sigma_h \subseteq \Sigma$. Let $H$ be a deterministic automaton having one state and with self-loops labeled by every event in $\Sigma_h$. The augmentation of $P$ by $\Sigma_h$, denoted by $P^{\Sigma_h}$, is defined to be the automaton $P^{\Sigma_h} := P_{\mathcal{A} \mathcal{B}} H$, with $\Sigma_v = \Sigma_h$, and the remaining 15 sets of PCXG as empty sets.

The state space of $P^{\Sigma_h}$ is identical to the state space of $P$, and $P^{\Sigma_h}$ is obtained by adding self-loops at each state on all events in $\Sigma_h$ except those already defined at that state.

5.2 PCXG based Decision Fusion

The decentralized supervisory control architecture of proposed in [19], and shown in Figure 7, uses an AND/OR logic for fusing decisions of local supervisors to obtain a global decision. The local supervisors to be combined agree a priori whether a controllable event belongs either to the AND set, so that the event is enabled when all the local supervisors permit it to occur, or to the OR set, so that the event is enabled when at least one of the local supervisors permit it to occur. Individual local supervisors can observe the plant events only partially and make decision fusion decisions based on the masked values of the event trace $s$ generated by the plant, that is observed as $M_i(s)$ by the $i^{th}$ local supervisors.

This division of the controllable event set into two, AND/OR, can further be extended to:

1. four divisions, if the partitions of PSC are used,
2. eight divisions, if the partitions of PCX are used, and
3. sixteen divisions, if the partitions of PCXG are used.
A block diagram of the most general decision fusion architecture, possible through PCXG is given in Figure 8. This architecture allows for the most general fusion of local decisions, and generalizes the one based on AND/OR fusion proposed in [19]. Results on decentralized supervisory control utilizing a PCXG decision fusion architecture will be reported in a future paper. It should be noted that since each boolean function can be represented using a pair of boolean operations, say AND and NOT, it is possible to represent each fusion block in Figure 8 using a boolean circuit of the pair of chosen boolean functions, such as OR(.,.) equals NOT(AND(NOT(.,),NOT(.,))).

6 PCXG OF THE RULES BASED MODEL

In Section 5 we defined PCXG of DESs modeled using automata. The definition can be extended to DESs modeled in the rules based modeling formalism [5]. In the rules based modeling formalism, the DES model of a system consists of an initial condition, \( I \), and occurrence rules for events, \( \{G_{\sigma} \Rightarrow \sigma \mid \sigma \in \Sigma\} \).
Here $I$ represents the initial state predicate, and $G_\sigma$ represents the enabling predicate guard for event $\sigma \in \Sigma$.

**Definition 6** PCXG of the Rules based Model:

Given two rules based models indexed by $i = \{1, 2\}$, having initial condition $I_i$, event occurrence rules \{\(G_\sigma^i \Rightarrow \sigma \mid \sigma \in \Sigma\)\}, and a partition of the events set, then the initial condition for the composed system is $I_1 \land I_2$, and the event occurrence rule for an event $\sigma \in \Sigma_{OP}$, where $\Sigma_{OP}$ is a partition of the event set, is given by

\[
(G_\sigma^1 \circ \Sigma_{OP}) (G_\sigma^2) \Rightarrow \sigma.
\]

**Example 4** Rules based model of a process control system:

Consider the various tank system configurations shown in Figure 9 where two taps $t_1$ and $t_2$ with equal flow rates are used under different configurations for filling the tank, whose level is monitored by a level sensor $n$. The signals in the system are $t_1$, $t_2$, $n$, and the events that can occur in this system are $t_{1on}$, $t_{1off}$, $t_{2on}$, $t_{2off}$, $n_{up}$, $n_{dn}$. Assume that the filling rates of both taps $t_1$, and $t_2$ as $+1$.

The manner in which the filling taps are connected affects the filling ($n_{up}$) event of the tank. Consider first the effect of just a single tap, $t_i$, (either $t_1$ or $t_2$)
COMPOSITION OF DISCRETE EVENT SYSTEMS

Fig. 9. Tank configurations

\( t_1 \text{ and } t_2 \text{ connected in an AND filling configuration} \) (a) \( \quad \)
\( t_1 \text{ and } t_2 \text{ connected in an OR filling configuration} \) (b)

\( t_2 \text{ connected in reverse - } t_2 \text{ connected in spare loop and does not effect flow in tank} \) (c)

\( t_2, \) on the filling of the tank. The rules based model of just a single filling tap and associated level sensor is given in Figure 10.

\[ \text{Fig. 10. Rules based model of the tank system with single filling tap } t_i \]

1. Initial conditions: \( t_i = [\text{off}] \); \( n = [\text{down}] \).
2. Event occurrence rules:

\[ \text{Rule}_{1}^{up}: [t_i^{+1}] \land [n] \Rightarrow nup; \quad \text{Rule}_{1}^{dn}: \text{false} \Rightarrow ndn; \]
\[ \text{Rule}_{2}^{on}: \overline{t_i} \Rightarrow tioff; \quad \text{Rule}_{2}^{off}: t_i \Rightarrow tioff. \]

\[ \text{Fig. 10. Rules based model of the tank system with single filling tap } t_i \]

**Example 5** \( \text{PCXG of the rules based model:} \)

For each of the configurations shown in Figure 9 the rules based model for the entire system are obtained by applying PCXG of the individual models as follows:

- The set partitions under PCXG for the configuration of Figure 9(a) are given by: \( \Sigma_\Lambda = \{t1up, t1dn, t2up, t2dn, nup, ndn\} \). This is because the physical positioning of the taps and sensor elements requires that all events in the two systems be synchronized. Further, since only the rule for \( nup \) contains a mixture of both actuator and sensor signals only it is
altered in the composed model, while the rules for \( t1up, t1dn, t2up, t2dn \), and \( ndn \) remain unchanged. The resulting event occurrence rule for \( nup \) becomes:

\[
Rule_{up}^{1}: [t1^{+1} \land t2^{+1}] \land [\overline{n}] \Rightarrow nup.
\]

- The set partitions under PCXG for the configuration of Figure 9(b) are given by: \( \Sigma_{\lambda} = \{t1up, t1dn, t2up, t2dn\} \), \( \Sigma_{\nu} = \{nup, ndn\} \). Filling the tank, i.e., the event \( nup \) can occur when either or both of the taps \( t1 \) or \( t2 \) are on. The occurrence of actuator events \( t1up, t1dn, t2up, \) and \( t2dn \) has to be synchronized as before between the two models. Thus the resulting event occurrence rule for \( nup \) becomes:

\[
Rule_{up}^{1}: [t1^{+1} \lor t2^{+1}] \land [\overline{n}] \Rightarrow nup.
\]

- The set partitions under PCXG for the configuration of Figure 9(c) are given by: \( \Sigma_{\lambda} = \{t1up, t1dn, t2up, t2dn\} \), \( \Sigma_{\nu} = \{nup, ndn\} \). Since the tap \( t2 \) has been placed in a loop placed series with tap \( t1 \), it does not affect the filling of the tank. The occurrence of actuator events \( t1up, t1dn, t2up, \) and \( t2dn \) has to be synchronized as before between the two models. Thus the resulting event occurrence rule for \( nup \) becomes: \( Rule_{up}^{1} : [(t1^{+1} \land t2^{+1}) \lor (t1^{+1} \land \overline{t2})] \land [\overline{n}] \Rightarrow nup \). Alternately,

\[
Rule_{up}^{1} : [t1^{+1}] \land [\overline{n}] \Rightarrow nup.
\]

- The set partitions under PCXG for the configuration of Figure 9(d) are given by: \( \Sigma_{\lambda} = \{t1up, t1dn, t2up, t2dn\} \), \( \Sigma_{\nu} = \{nup, ndn\} \). This is because the tap \( t2 \) is reverse connected, permitting flow to occur through itself when it is turned off, and vice versa. It is also in series with tap \( t1 \), and hence filling can occur only when \( t1 \) is on, and \( t2 \) is off. Thus the resulting event occurrence rule for \( nup \) becomes:

\[
Rule_{up}^{1} : [t1^{+1} \land (\overline{t2})^{+1}] \land [\overline{n}] \Rightarrow nup.
\]

7 CONCLUSIONS AND FUTURE WORK

The notion of prioritized synchronous composition proposed by Heymann [7] is extended to include additional operations under which the composition of two automata may be performed. Two new interaction mechanisms for automata, Prioritized Composition with Exclusion (PCX), and Prioritized Composition with Exclusion and Generation (PCXG) have been proposed, which make modeling complex interactions among automata based models more general.
Under PSC, when two automata $P$ and $Q$ interact using priority sets $A$ and $B$ respectively, the events in the priority sets specify the events in whose occurrence the respective automaton must participate. This interaction can equivalently be obtained by partitioning the entire event set of the two interacting automata $P$ and $Q$ into four partitions denoted as $\Sigma_A, \Sigma_B, \Sigma_p, \Sigma_q$. Under PCX, events can also occur when exactly one of the interacting automata permit the event to occur. This is accomplished by the introduction of an exclusivity set, $C$, in addition to the priority sets $A$ and $B$. Introduction of $C$ is equivalent to introducing four new set partitions, denoted as $\Sigma_{\oplus}, \Sigma_{\otimes}, \Sigma_{\lor}, \Sigma_{\land}$. Finally, under PCXG, events can also occur when neither of the interacting automata permit the event to occur. This is accomplished by the introduction of a nor-generative set, $D$. Introduction of $D$ is equivalent to introducing eight new set partitions, denoted as $\Sigma_{p+}, \Sigma_{q+}, \Sigma_{\land+}, \Sigma_{\lor+}, \Sigma_{\land+}, \Sigma_{\lor+}, \Sigma_{\land\lor+}, \Sigma_{\land\lor+}$.

Both PCX and PCXG interaction reduce to PSC, and SC under special conditions, and can thus be used to perform all the interactions supported by PSC and SC. The properties of the PCX and PCXG interaction mechanism has been studied. Applicability of these interactions to decentralized supervisory decision fusion, as well as to the rules based modeling formalism has been illustrated. Future extensions will study the properties of PCX and PCXG under partial observation, and its further application to the control of interactions.

ACKNOWLEDGEMENTS

The research was supported in part by National Science Foundation under the grants NSF-ECS-9709796, NSF-ECS-0099851, NSF-ECS-0218207, and NSF-ECS-0244732, a DoD-EPSCoR grant from the Office of Naval Research under the grant N000140110621, and a KYDEPSCoR grant.

REFERENCES


