Proceedings of the 9th International
Workshop on Discrete Event Systems
Göteborg, Sweden, May 28-30, 2008

Coordination Control of Discrete-Event Systems

Jan Komenda
Institute of Mathematics, Czech Academy of Sciences,
Brno Branch, Zizkova 22, 616 62 Brno, Czech Republic
komenda@ipm.cz

Jan H. van Schuppen
CWI, P.O. Box 94079,
1090 GB Amsterdam, The Netherlands
J.H.van.Schuppen@cwi.nl

***Abstract*— The concept of a *coordinator* is proposed for control of modular discrete-event systems. The coordinator makes all subsystems conditionally independent generators as defined in the paper. The coordinator receives part of the partial observations of the subsystems and its task is to satisfy the global part of the specification and of the nonblockingness. The complete supervisor then consists of the coordinator, its supervisor, and the local supervisors for the subsystems. An example of control of a distributed discrete-event system shows that a coordinator is necessary for achieving safety and nonblockingness.**

I. INTRODUCTION

The purpose of this paper is to present the coordination control approach for control of decentralized and modular discrete-event systems. Because of the complexity of control problems and because of the way large engineering systems are constructed, there is an increasing interest in control of decentralized and modular systems.

Our previous results in modular control of DES rely on necessary and sufficient conditions for local control synthesis to equal global control synthesis for both local and global specifications and for both complete and partial observations. Interestingly, we have shown recently in [7] that the structural conditions used for global specifications are equivalent to those used for local specification. However if these conditions are not satisfied, either the system must be modified to meet them or a hierarchical approach (as the one that was proposed in [11]) should be adopted such that the abstracted system meets these structural conditions. Moreover blocking issues have not been considered in our earlier approach.

Therefore we have developed an approach based on coordination to handle the blocking issues. Coordination of discrete-event systems was discussed earlier by Kai C. Wong and W. Murray Wonham in [14]. A similar approach to ours has also been proposed in [4], where the authors have shown that if all projections from local alphabets that abstract away events which are not included in the coordinator sets are observers of the local systems then there always exists a coordinator which makes the composed system nonblocking. The main difference is that our method provides both sufficient and necessary conditions on the coordinator to make the composed system nonblocking.

The research was supported by Grant Agency of Acad. of Sci. of Czech Republic and Acad. of Sci. of Czech Republic, Inst. Research Plan No. AV0Z10190503.

Coordination control is based on the concept of conditional independence of σ-algebras which is used in probability theory. The coordinator generator makes two or more generators conditionally independent if the joint action of two or more generators is always accompanied by a transition of the coordinator. Conditional independence can also be defined for automata languages. A coordinator always exists so the search is for the smallest possible coordinator. The novelty of the paper is in the concept of a coordinator for safety and for nonblockingness based on the concept of conditional independence. The approach is inspired by the work of K.C. Wong, W.M. Wonham, and coworkers on modular control, by research of the authors of this paper on modular control, and by publications of K. Schmidt and T. Moor on a hierarchical approach to control, see [12].

A description of the contents follows. The next section presents a problem formulation and motivation of the problem. In Section III the basic concepts of the paper: conditional independence of languages and of generators is introduced. In Section IV these concepts are studied and related to each other. Section V shows how nonblocking can be achieved by composing the modular plant with a suitable coordinator. In Section VI safety is studied. The control synthesis of the coordinator with supervisors is discussed in Section VI. The proofs are included in the appendix.

II. PROBLEM FORMULATION

The motivation for coordination control is decentralized and modular control of discrete-event systems (DES). Consider a modular DES. It is well known, see the lecture notes of W.M. Wonham, [15], that a supervisor for the modular system for which a controller has been synthesized using modular control, may be blocking. Example 5.5 in Section V establishes that there does not exist a set of local supervisors which is nonblocking unless the observations of both models are sent to a global coordinator who can then prevent blocking by disabling particular events. Thus the nonblocking property of modular control cannot always be achieved modularly and nonblockingness requires a coordinator at the global level. See for modular control of DES [8] and [2].

The main problem for both decentralized control and modular control is the construction of a coordinator. A coordinator always exists, a supervisor for the global plant will be a coordinator. The difficult part of the problem is to

978-1-4244-2593-8/08/$25.00 ©2008 IEEE

construct a minimal coordinator which is least restrictive in regard to the local subsystems.

Terminology and Notation

The terminology of DES is more or less according to the lecture notes of W.M. Wonham [15] and the book [1] but the notation of this paper differs slightly. A (deterministic) *generator*

$$G = (Q, E, f, q_0, Q_m), \tag{1}$$

is a mathematical structure with *state set* Q, an *event set* E, a *partial transition function* $f : Q \times E \to Q$, an *initial state* $q_0 \in Q$, and a *subset of marked states* $Q_m \subseteq Q$. A transition is also denoted as $q \overset{e}{\mapsto} q^+ = f(q,e)$. If a transition is defined then this is denoted by $f(q,e)!$ Extend the transition function f to $f : Q \times E^* \to Q$ by induction. Define respectively the *language* and the *marked language* of the generator as,

$$\begin{aligned} L(G) &= \{s \in E^* | f(q_0, s)!\}, & (2)\\ L_m(G) &= \{s \in L(G) | f(q_0, s) \in Q_m\}. & (3) \end{aligned}$$

A *controlled generator* is a structure (G, E_c, Γ_c), where G is a generator, $E_c \subseteq E$ is the subset of *controllable events*, $E_{uc} = E \backslash E_c$ is the subset of *uncontrollable events*, and $\Gamma_c = \{\gamma \subseteq E | E_{uc} \subseteq \gamma\}$, is called the *set of control patterns*. A *supervisor* for the controlled generator is map $g : L(G) \to \Gamma_c$. The *closed-loop system* associated with a controlled generator and a supervisor as denoted above is defined as the smallest language $L(S/G) \subseteq E^*$ and the marked language $L_m(S/G) \subseteq L(S/G)$ which satisfy respectively,

$$\begin{aligned} &(1) && \varepsilon \in L(S/G),\\ &(2) && \text{if } s \in L(S/G),\ se \in L(G) \text{ and if } e \in g(s)\\ & && \text{then } se \in L(S/G).\\ L_m(S/G) &= && L(S/G) \cap L_m(G). \end{aligned}$$

Recall that the *natural projection* $P : E^* \to E_o^*$ is a morphism of monoids such that $P(\varepsilon) = \varepsilon$ and P erases the events that are not in $E_o \subseteq E$. A supervisor with partial observations is a map $g : P(L(G)) \to \Gamma_c$.

It is important to distinguish for a generator between an event set and its associated reachable event set. Note that there may exist events of the event set which do not appear in any transition. Moreover, even if an event is used for a transition then that transition may not be reachable. This applies in particular to the case of the synchronous product of two generators.

Definition 2.1: Consider a generator G denoted as above. Define the subset of *reachable events*, denoted by $E_r(G) \subseteq E$, if for any event $e \in E_r(G)$ there exists a string $s \in E^*$ containing the event e for which the function $f(q_0, s)$ is defined. Similarly, define for any language $L \subseteq E^*$ the subset of *reachable events* $E_r(L) \subseteq E$ *of the language* as the subset of events which occur in the strings of the language.

Note the abuse of notation in $E_r(G)$ and $E_r(L)$. The complexity of computing the event set $E_r(G)$ is $O(n(G) \times m_E(G))$ where $n(G)$ denotes the number of states and $m_E(G)$ denotes the number of events of the generator G.

Definition 2.2: Consider two event sets E_1 and E_2 and two languages $L_1 \subseteq E_1^*$ and $L_2 \subseteq E_2^*$. The *synchronous product* of the languages L_1 and L_2 is defined as

$$L_1 \| L_2 = P_1^{-1}(L_1) \cap P_2^{-1}(L_2).$$

where $P_i : (E_1 \cup E_2)^* \to E_i^*$ for $i = 1, 2$. Their synchronous product is called the *shuffle product* if

$$\begin{aligned} \emptyset &= E_{r,sh} = E_r(L_1 \| L_2) \cap E_1 \cap E_2, & (4)\\ &\text{and then one writes}\\ \text{shuffle}(L_1, L_2) &= L_1 \| L_2 = P_1^{-1}(L_1) \cap P_2^{-1}(L_2). & (5) \end{aligned}$$

The subset of reachable shared events $E_{r,sh}$ can be empty while $E_1 \cap E_2$ is not as a simple example shows. The corresponding synchronous product of generators [15] is known to satisfy

$$\begin{aligned} L(G_1 \| G_2) &= L(G_1) || L(G_2), & (6)\\ L_m(G_1 \| G_2) &= L_m(G_1) || L_m(G_2). & (7) \end{aligned}$$

A distributed discrete-event system is a modular or a concurrent system with the global plant formed by the synchronous product of local subsystems.

Definition 2.3: A *modular discrete-event system* with two modules is a structure $(G_1, G_2, E_{1,c}, \Gamma_{1,c}, E_{2,c}, \Gamma_{2,c})$ consisting of two modules in the form of controlled generators. The associated global system is their synchronous product $G_1 \| G_2$. Denote the natural projections by

$$P_1 : (E_1 \cup E_2)^* \to E_1^*,\ P_2 : (E_1 \cup E_2)^* \to E_2^*.$$

Throughout the paper the special case of two modules is considered in order to simplify the exposition. A coordinator of a modular system will be illustrated in Example 5.5.

III. CONCEPTS

In this section the concept of conditionally independent generators and related notions are defined.

Conditional independence of σ-algebras is a concept of probability theory which has been used to put the concept of state of a stochastic system on a fundamental basis, see [9] and the references quoted there. A corresponding notion is useful in automata theory as well. This section presents the concepts, coordination control theory with these concepts is presented in the following sections.

Denote $E = E_1 \cup E_2$ and $E_k \subseteq E$ a coordinator alphabet. The following natural projections are needed: $P_{1\cup k} : E^* \to (E_1 \cup E_k)^*$, $P_1^{1\cup k} : (E_1 \cup E_k)^* \to E_1^*$, and the corresponding inverse projection $(P_1^{1\cup k})^{-1} : \text{Pwr}(E_1^*) \to \text{Pwr}(E_1 \cup E_c)^*$. Similarly, $P_{2\cup k} : E^* \to (E_2 \cup E_k)^*$, $P_2^{2\cup k} : (E_2 \cup E_k)^* \to E_2^*$, and $(P_2^{2\cup k})^{-1} : \text{Pwr}(E_2^*) \to \text{Pwr}(E_2 \cup E_k)^*$ are defined. Symmetrically, let $P_k^{i\cup k} : (E_i \cup E_k)^* \to E_k^*$, $i = 1, 2$. Also, let $P_{i\backslash k} : E^* \to (E_i \backslash E_k)^*$, $i = 1, 2$. The notation $P_{i\cap k}^i : E_i^* \to (E_i \cap E_k)^*$, $i = 1, 2$ is now self-explanatory.

Definition 3.1: Consider three generators,

$$\begin{aligned} G_k &= (Q_k, E_k, f_k, q_{k,0}, Q_{k,m}), & (8)\\ G_1 &= (Q_1, E_1, f_1, q_{1,0}, Q_{1,m}), & (9)\\ G_2 &= (Q_2, E_2, f_2, q_{2,0}, Q_{2,m}). & (10) \end{aligned}$$

Call G_1, G_2 *conditionally independent generators* given G_k if for any reachable state in the synchronous product $(q_k, q_1, q_2) \in G_k \| G_1 \| G_2$, there does not exist a transition of the form,

$$(q_k, q_1, q_2) \mapsto (q_k, q_1^+, q_2^+), \tag{11}$$

and $a \notin E_k$

Note that conditional independence means that there is no simultaneous move in both G_1 and G_2 without the coordinator being also involved. The concept is easily extended to the case of three or more generators. The corresponding concept in terms of languages follows.

Definition 3.2: Consider event sets E_1, E_2, E_k and languages $L_1 \subseteq E_1^*$, $L_2 \subseteq E_2^*$, and $L_k \subseteq E_k^*$. The languages L_1, L_2 are said to be *conditionally independent given* L_k if $E_r(L_1 \| L_2) \cap E_1 \cap E_2 \subseteq E_k$.

Notation. $(L_1, L_2 | L_k) \in \text{CIL}$ denotes that the languages L_1, L_2 are conditionally independent given L_k.

Other related concepts are defined below.

Definition 3.3: Consider the events sets E_1, E_2 and E_k and the languages $L_1 \subseteq E_1^*$, $L_2 \subseteq E_2^*$, $L_k \subseteq E_k^*$, and $K \subseteq E^*$. Assume that $E_r(L_1 \| L_2) \cap E_1 \cap E_2 \subseteq E_k \subseteq E_1 \cup E_2 = E$. Define the conditions:

(a) The triple of languages (L_1, L_2, L_k) is called *conditionally shuffle closed* if

$$\begin{aligned} & t \in L_1 \| L_2 \| L_k, \text{ with decomposition,} \\ t \quad = \quad & s_1 t_{c1} \ldots s_k t_{ck} s_{k+1}, \\ & s_1, \ldots s_{k+1} \in [(E_1 \backslash E_k) \cup (E_2 \backslash E_k)]^*, \\ & t_{c,1}, \ldots, t_{c,k} \in E_k^* \\ \text{PS}(s_i) \quad \in \quad & \text{shuffle}(P_{1\backslash k}(s_i), P_{2\backslash k}(s_i)), \ \forall i = 1, \ldots, k+1, \\ \bar{t} \quad \in \quad & \text{PS}(s_1) t_{c1} \ldots \text{PS}(s_k) t_{c,k} \text{PS}(s_{k+1}) \\ \Rightarrow \quad & \bar{t} \in L_1 \| L_2 \| L_k. \end{aligned}$$

(b) The triple of languages (L_1, L_2, L_k) is called *conditionally projection-closed* if for all $t, \bar{t} \in E^*$, $t \in L_1 \| L_2 \| L_k$, $P_k(t) = P_k(\bar{t})$, $P_{1\cup k}(t) = P_{1\cup k}(\bar{t})$, and $P_{2\cup k}(t) = P_{2\cup k}(\bar{t})$ imply that $\bar{t} \in L_1 \| L_2 \| L_k$.

(c) The language K is called *conditionally decomposable* with respect to the event sets (E_1, E_2, E_k) if

$$K = P_{1\cup k}(K) \| P_{2\cup k}(K) \| P_k(K).$$

The term conditionally decomposable is used, because the projections $P_{i\cup k} : E^* \to E_{i\cup k}^*$ are involved. It should be clear that conditional decomposability is weaker than decomposability with respect to $P_1 : E^* \to E_1^*$, $P_2 : E^* \to E_2^*$, and $P_k : E^* \to E_k^*$ as defined in the literature on decentralized control. This is because $E_i \subseteq E_i \cup E_k$ implies that $P_i^{-1} P_i(K) \subseteq P_{i\cup k}^{-1} P_{i\cup k}(K)$. Also note that in case $E_k = E = E_1 \cup E_2$ conditional decomposability is trivially satisfied.

IV. Equivalent conditions for conditional independence

The concepts of conditional indepence and conditional decomposability are closely related. First of all we notice from the very definitions that the following Proposition holds.

Proposition 4.1: Consider the generators of Definition 3.1. The following statements are equivalent:

(a) G_1 and G_2 are conditionally independent given G_k.
(b) The languages $(L(G_1), L(G_2) | L(G_k)) \in \text{CIL}$ are conditionally independent.
(c) $E_r(G_1 \| G_2) \cap E_1 \cap E_2 \subseteq E_k$.

Theorem 4.2: Consider the events sets E_1, E_2 and E_k and the languages $L_1 \subseteq E_1^*$, $L_2 \subseteq E_2^*$, and $L_k \subseteq E_k^*$.

(a) If the languages L_1, L_2 are conditionally independent given L_k then the triple (L_1, L_2, L_k) is conditionally shuffle closed.
(b) Assume that the languages L_1, L_2 are conditionally independent given L_k. Then the following statements are equivalent:
  (b.1) The triple (L_1, L_2, L_k) is conditionally shuffle closed.
  (b.2) The triple (L_1, L_2, L_k) is conditionally projection closed.
  (b.3) The language $L_1 \| L_2 \| L_k$ is conditionally decomposable.

The technical results below will be needed in the remaining sections.

Lemma 4.3: [3, Proposition 4.3] Let $E_1 \cap E_2 \subseteq E_k$ and $L_i \subseteq E_i^*$, $i = 1, 2$. Then $P_k(L_1 \| L_2) = P_{1\cap k}^1(L_1) \| P_{2\cap k}^2(L_2)$.

Let us note that the condition $E_1 \cap E_2 \subseteq E_k$ in the above lemma can be weakened to our condition $E_r(G_1 \| G_2) \cap E_1 \cap E_2 \subseteq E_k$: some of the events from $E_1 \cap E_2$ might actually not be reachable in $G_1 \| G_2$ and these need not be included in E_k for the lemma to hold true.

Lemma 4.4: Let $L_i \subseteq E_i^*$, $i = 1, 2, k$ and $E_r(G_1 \| G_2) \cap E_1 \cap E_2 \subseteq E_r(G_k)$. Then

(1) $P_k(L_1 \| L_2 \| L_k) \subseteq L_k$
(2) $P_{1\cup k}(L_1 \| L_2 \| L_k) \subseteq L_1 \| L_k$
(3) $P_{2\cup k}(L_1 \| L_2 \| L_k) \subseteq L_2 \| L_k$

V. Coordinator for nonblockingness

In this section the nonblockingness of the complete system is studied. The complete system consists of the coordinator in parallel composition with the two modules. In the following definition the equivalent condition of nonblockingness is stated. Consider again two generators and their coordinator,

$$G_1 = (Q_1, E_1, f_1, q_{1,0}, Q_{1,m}), \tag{12}$$
$$G_2 = (Q_2, E_2, f_2, q_{2,0}, Q_{2,m}), \tag{13}$$
$$G_k = (Q_k, E_k, f_k, q_{k,0}, Q_{k,m}). \tag{14}$$

Definition 5.1: We say that $G = G_1 \| G_2$ is *conditional nonblocking* given the coordinator automaton G_k if

(1) $\forall s \in \bar{L}_m(G_1) \| \bar{L}_m(G_2) \| \bar{L}_m(G_k) \quad \exists t_k \in E_k^* : P_k(s) t_k \in L_m(G_k)$ and
(2) (i) conditional nonblockingness of $G_1 \| G_k$ holds, i.e. $\exists v \in (E_1 \cup E_k)^*$ such that $P_1(s) P_1^{1\cup k}(v) \in L_m(G_1)$ and $P_k^{1\cup k}(v) = t_k$ and
(ii) conditional nonblockingness of $G_2 \| G_k$ holds, i.e. $\exists w \in (E_2 \cup E_k)^*$ such that $P_2(s) P_2^{2\cup k}(w) \in L_m(G_2)$ and $P_k^{2\cup k}(w) = t_k$.

The conditional nonblockingness represents a good compromise between existence of local prolongations to local marked states (i.e. local nonblocking) and the existence of global prolongations to marked states (i.e. global nonblocking). It is well known that local nonblocking is not enough to ensure the global nonblocking. Therefore a coordinator layer is added to the modular system such that the coordinator together with the first subsystem as well as coordinator with the second subsystem are nonblocking in the sense of the definition 5.1. The following theorem then states that the conditional nonblockingness defined above is not only sufficient, but also necessary for the nonblockingness of the composed system $G_k \| G_1 \| G_2$.

Theorem 5.2: Consider the setting of Definition 5.1. Assume that the coordinator G_k makes G_1 and G_2 conditionally independent, i.e. G_1, G_2 are conditionally independent generators given G_k in the sense of Definition 3.1. The composed system $G_1 \| G_2 \| G_k$ is nonblocking if and only if G is conditional nonblocking given G_k.

In the last theorem the nonblocking coordinator, which makes the resulting system nonblocking, is characterized using the conditions (2) (i) and (ii) of Def. 5.1. The first question is whether this characterization (in terms of strings) can be verified. Next a procedure will be given (with immediate automata interpretation), which checks whether conditions (1) and (2): (i) and (ii) above are satisfied. The key steps will be computations of coreachable sets and projected generators (automata) which can be computed using a standard subset construction. Let us recall that inverse projections of automata are obtained for free, just by adding selfloops of events not observable with respect to the corresponding projection.

We show that the candidate sets for the strings t_k, v, and w obeying the properties (i) and (ii) can be found in the following way. For any $s \in \overline{L}_m(G_1) \| \overline{L}_m(G_2) \| \overline{L}_m(G_k)$ we have $P_i(s) \in \overline{L}_m(G_i)$, $i = 1,2$ and $P_k(s) \in \overline{L}_m(G_k)$. We look for extensions of $P_i(s)$ $i = 1,2,k$ within $L_m(G_i)$, $i = 1,2,k$, i.e.

$$S_i(s) := \{t_i \in E_i^* : P_i(s)t_i \in L_m(G_i))\}.$$

These languages can be easily computed using coreachable sets in their corresponding automata: CoReachset$(Q_{i,m}, E_i^*)$ and intersecting them with the quotient language of G_i by $P_i(s)$, $i = 1,2,k$ represented by states in the automata G_i after the strings $P_i(s) \in E_i^*$, $i = 1,2,k$ have been generated. Hence, $S_i(s)$, $i = 1,2,k$ are fairly easily computable.

Now, the candidates for the strings t_k, v, and w of Definition 5.1 can be found in terms of $S_i(s)$, $i = 1,2$ and suitable projections and inverse projections. Hence, we consider the intersection $P_{1\cup k}^{-1}(S_1) \cap P_{2\cup k}^{-1}(S_2)$, which can be implemented by the standard automata composition. The following simple fact is needed.

Property 5.3: For languages $L_i \subseteq E^*$, $i = 1,2$ and natural projection $P_k : E^* \to E_o^*$ with $E_o \subseteq E$ we have $s \in P(L_1) \cap L_2 \neq \emptyset \Rightarrow P^{-1}(s) \subseteq L_1 \cap P^{-1}(L_2) \neq \emptyset$.

Proof: Let $s \in P(L_1) \cap L_2 \neq \emptyset$. Then $s \in L_2$ and there exists $t \in L_1$ such that $s = P(t)$. Hence, $t = P^{-1}(s) \subseteq P^{-1}(L_2)$, i.e. $t \in L_1 \cap P^{-1}(L_2) \neq \emptyset$. □

We obtain a computable criterion for checking the necessary and sufficient condition of Theorem 5.2.

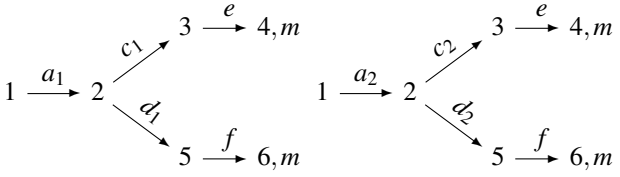
Proposition 5.4: $G_1 \| G_2$ is conditional nonblocking given the coordinator automaton G_k if and only if

$$\begin{aligned} &\forall\ s \in \overline{L}_m(G_1) \| \overline{L}_m(G_2) \| \overline{L}_m(G_k), \\ &[P_k^{1\cup k}(P_1^{1\cup k})^{-1}(S_1(s)) \cap S_k(s)] \cap \\ &\cap [P_k^{2\cup k}(P_2^{2\cup k})^{-1}(S_2(s)) \cap S_k(s)] \neq \emptyset. \end{aligned} \tag{15}$$

It is important that the criterion of the Proposition 5.4 is checkable, because $S_i(s)$, $i = 1,2,k$ can be constructed using coreachable sets and all projections, inverse projections and intersections can be computed as has been pointed out above.

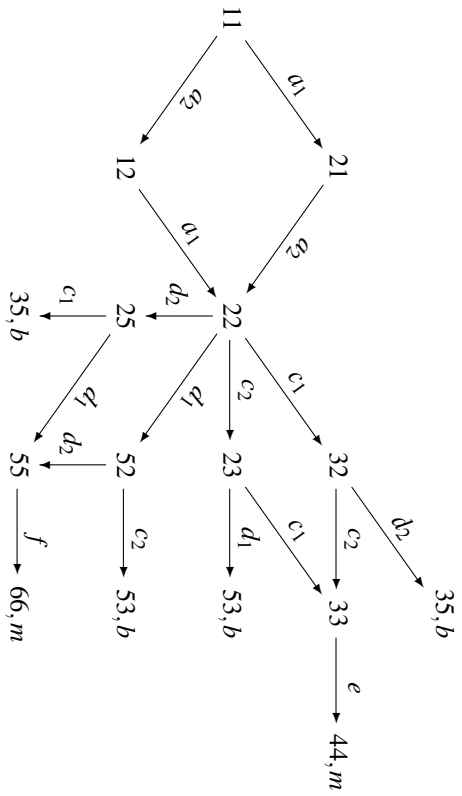
Let us now illustrate the above decribed procedure and action of the coordinator on the example 5.5.

Example 5.5: Let us consider the following local plant languages.

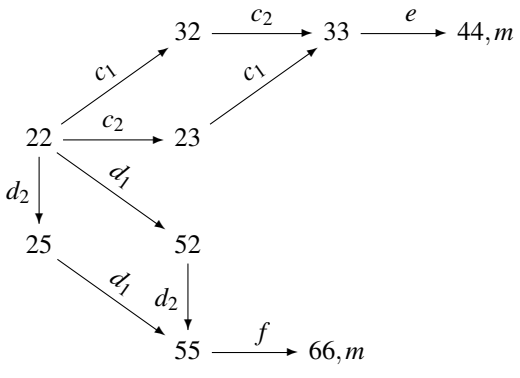


$$\begin{aligned} &E_1 = \{a_1, c_1, d_1, e, f\},\ E_2 = \{a2, c_2, d_2, e, f\}, \\ &E_k = \{e, f, c_1, c_2, d_1, d_2\} \supseteq E_1 \cap E_2 = \{e, f\}, \\ &G_1 \| G_2,\ \text{blocking}. \end{aligned}$$

We assume that the subset of marked states are $Q_{1m} = \{4,6\}$ and $Q_{2m} = \{4,6\}$, which are denoted by small m. It is easily seen that the synchronous product of these local plants involve blocking.



It can be easily seen that not all private events have to be included into the coordinator (high level) event set E_k. Indeed, a_1 and a_2 need not be included in E_k. The choice $E_k = E \setminus \{a_1, a_2\}$ is optimal for the composed system. Indeed, the following choice of G_k is optimal such that the composed system $G_1 \| G_2 \| G_k$ is nonblocking.



Let us also illustrate on this example the presentation of Proposition 5.4. Consider the string $s = a_1 a_2 c_1$. Then $P_1(s) = a_1 c_1$ and $P_k(s) = c_1$. Furthermore, $S_k(s) = \{t \in E_k^* \mid c_1 t \in L_m(G_k)\} = \{c_2 e\}$, $S_1(s) = \{t \in E_1^* \mid a_1 c_1 t \in L_m(G_1)\} = \{e\}$. Hence, $(P_1^{1\cup k})^{-1}(S_1(s)) = (c_2^* d_2^*)^* e (c_2^* d_2^*)^*$ and $P_k^{1\cup k}(P_1^{1\cup k})^{-1}(S_1(s)) = \{c_2 e\}$. Similarly, $P_2(s) = a_2$, i.e. $S_2(s) = \{t \in E_2^* \mid a_2 t \in L_m(G_2)\} = \{c_2 e, d_2 f\}$ and $P_k^{2\cup k}(P_2^{2\cup k})^{-1}(S_2(s)) = (c_1^* d_1^*)^* c_2 (c_1^* d_1^*)^* e (c_1^* d_1^*)^* \cup (c_1^* d_1^*)^* d_2 (c_1^* d_1^*)^* f (c_1^* d_1^*)^*$. Therefore, $P_k^{2\cup k}(P_2^{2\cup k})^{-1}(S_2(s)) \cap S_k(s) = \{c_2, e\}$. It is sufficient to take $t_k = c_2 e$ and the corresponding strings v and w satisfying (i) and (ii) of Definition 5.1, which are $v = c_2 e$ and $w = c_2 e$. The role of the coordinator is to allow after the string $s = a_1 a_2 c_1$, i.e. in the corresponding state of G_2, only the string c_2, which can be then extended to the marked string $a_1 a_2 c_1 c_2 e$, while disable string d_2, which leads to blocking.

The conclusion is then $P_k^{1\cup k}(P_1^{1\cup k})^{-1}(S_1(s)) \cap P_k^{2\cup k}(P_2^{2\cup k})^{-1}(S_2(s)) \cap S_k(s) \neq \emptyset$. This can be shown for any other strings $s \in \overline{L_m}(G_1) \| \overline{L_m}(G_2) \| \overline{L_m}(G_k)$, which means by Proposition 5.4 and Theorem 5.2 that $G = G_1 \| G_2$ is conditional nonblocking given G_k.

Yet another important problem is the construction of the coordinator, i.e. how the coordinator can actually be synthesized in an automated manner. First of all, such a coordinator clearly exists.

Proposition 5.6: Consider a modular DES as in Definition 2.3. There exists a generator G_k, denoted as in Definition 3.1, such that G_1, G_2 are conditionally independent generators given G_k. In fact, $G_k = trim(G_1 \| G_2)$, i.e. the part of the global system that is both reachable and coreachable, certainly works as a coordinator for nonblockingness.

The results of [3] are helpful. The question is whether/how our condition is related to the observer property from [13]. In particular, whether there is a sufficient condition for modular nonblockingness in terms of the observer properties of suitable projections. Let us recall first the definition of an observer.

Definition 5.7: For a language $L \subseteq E^*$, the natural projection $P : L \to P(L) \subseteq E_o^*$ is said to be an *L-observer* if for any $t \in P(L)$ and any $s \in \bar{L}$ such that $P(s) \leq t$ there exists $u \in E^*$ with the properties $su \in L$ and $t = P(su)$.

This means that for any abstracted extension within the abstracted system's language there must exist a compatible extension within the language of the "original" system, whence the idea to adapt the observer property used in hierarchical systems for use in modular and distributed systems. If $P_{i\cap c}^i$ is a L_i-observer for $i = 1,2$ then we know from [3, Proposition 4.10] that there always exist a coordinator. Namely, $G_k := P_{i\cap c}^i(L_1) \| P_{i\cap c}^i(L_2)$ is a good choice of coordinator for nonblockingness of the composed system $G_1 \| G_2 \| G_k$.

Since the above condition is not necessary (unlike our characterizations (i) and (ii)), it follows that our conditions are weaker. On the other hand if $P_{i\cap k}^i$ are L_i-observers for $i = 1,2$ then our conditions as necessary and sufficient for nonblocking must be satisfied.

VI. Verification for safety

The purpose of this subsection is to present a condition for safety of concurrent discrete-event systems. We are interested in safety of modular systems composed with the coordinator. It will be assumed that a closed-loop system is specified both for the coordinator and for the remaining parts of the subsystems after the coordinator is imposed. Control synthesis of the coordinator and the supervisors will be discussed in section VII.

An equivalent condition for verification of safety is conditional safety defined below as safety of the coordinator

and safety of each local subsystem when combined with the coordinator.

Problem 6.1: Consider two generators G_1, G_2, and a coordinator G_k, which makes G_1 and G_2 conditionally independent. Consider a specification language $K \subseteq E^*$. Assume that the language K is conditionally decomposable with respect to the reachable event sets (E_k, E_1, E_2). Determine sufficient or equivalent conditions such that
$L_m(G_1)\|L_m(G_2)\|L_m(G_k) \subseteq K$.

We have first the following result.

Proposition 6.2: Let $K \subseteq E^*$ be conditionally decomposable with respect to the reachable event sets (E_k, E_1, E_2) and let there exist a coordinator G_k over E_k^* such that

(i) $L_m(G_k) \subseteq P_k(K)$
(ii) $L_m(G_1)\|L_m(G_k) \subseteq P_{1\cup k}(K)$
(iii) $L_m(G_2)\|L_m(G_k) \subseteq P_{2\cup k}(K)$

Then $L_m(G_1)\|L_m(G_2)\|L_m(G_k) \subseteq K$.

Now we give weaker (string based) necessary and sufficient conditions for safety similar to conditions used for nonblockingness.

Definition 6.3: Consider the setting of Problem 6.1. The system and the specification language $K \subseteq E^*$ are said to be *conditionally safe* with respect to the reachable event sets (E_k, E_1, E_2) if

(1) $L_m(G_k) \subseteq P_k(K)$
(2) $\forall t_k \in L_m(G_k)$:
$v \in L_m(G_1)\|L_m(G_k)$, $P_k^{1\cup k}(v) = t_k$, $w \in L_m(G_2)\|L_m(G_k)$, $P_k^{2\cup k}(w) = t_k \Rightarrow v \in P_{1\cup k}(K)$ and $w \in P_{2\cup k}(K)$.

Note that conditional safety is weaker than (i)-(iii) of Proposition 6.2. Indeed, while (i) and (1) are the same, (2) of Definition 6.3 is weaker than (ii) and (iii) of Proposition 6.2. This is because the assumptions of the implication are stronger than in the implications corresponding to (ii) and (iii) of Proposition 6.2, i.e. $v \in L_m(G_1)\|L_m(G_k) \Rightarrow v \in P_{1\cup k}(K)$ and idem for (iii) separately. Note that (2) is equivalent to the following inclusions:
2(i) $L_m(G_1)\|L_m(G_k) \cap (P_k^{1\cup k})^{-1} P_k^{2\cup k}(L_m(G_2)\|L_m(G_k)) \subseteq P_{1\cup k}(K)$
2(ii) $L_m(G_2)\|L_m(G_k) \cap (P_k^{2\cup k})^{-1} P_k^{1\cup k}(L_m(G_1)\|L_m(G_k)) \subseteq P_{2\cup k}(K)$.

The main result of this section is now formulated below.

Theorem 6.4: Consider Problem 6.1. If the system and the specification language be conditionally safe with respect to the reachable event sets (E_k, E_1, E_2) then $L_m(G_1)\|L_m(G_2)\|L_m(G_k) \subseteq K$. Conversely, if $L_m(G_1)\|L_m(G_2)\|L_m(G_k) \subseteq K$ and $L_m(G_k) \subseteq P_k(L_m(G_1)\|L_m(G_2))$, then the specification language are conditionally safe with respect to the reachable event sets (E_k, E_1, E_2).

Let us remark that $L_m(G_k) \subseteq P_k(L_m(G_1)\|L_m(G_2))$ is often satisfied in coordination control, because coordinators for safety as well as coordinators for nonblockingness typically do not add additional behavior to the composed systems, i.e. $L_m(G_k)$ is included in the projected behavior.

VII. Control synthesis

In this section the overall control synthesis is presented. Using the coordination scheme, first a supervisor for coordinator is synthesized that takes care of the part $P_k(K)$ of the specification K. Then S_i, $i = 1,2$ are synthesized such that the remaining part of the specification, i.e. $P_{i\cup k}(K)$ are met by the new plant languages $G_i\|(S_k/G_k)$. Let $E_u \subseteq E$ be the set of uncontrollable events and $E_{i,u} = E_u \cap E_i$, $i = 1,2,k$ the corresponding sets of local uncontrollable events.

Problem 7.1: Consider generators G_1, G_2, G_k and a specification language $K \subseteq (E_1 \cup E_2 \cup E_k)^*$. Assume that the coordinator G_k makes the two generators G_1, G_2 conditionally independent and that the language $\overline{K}$ is conditionally decomposable.

Determine supervisors S_1, S_2, S_k for the respective generators such that the closed-loop system with S_k/G_k as coordinator for S_1/G_1 and S_2/G_2 is such that

$$L(S_1/[G_1\|(S_k/G_k)])\|L(S_2/[G_2\|(S_k/G_k)])\|L(S_k/G_k) = \overline{K} \quad (16)$$

Definition 7.2: Consider the setting of Problem 7.1 Call the specification language $K \subset E^*$ *conditionally controllable* for generators (G_1, G_2, G_k) and for the event subsets $(E_{1,u}, E_{2,u}, E_{k,u})$ if

1) The language $P_k(K) \subseteq E_k^*$ is controllable with respect to G_k and $E_{k,u}$; equivalently,
$$\overline{P_k(K)}E_{k,u} \cap L(G_k) \subseteq \overline{P_k(K)}. \quad (17)$$
Then there then exists a nonblocking supervisor S_k for G_k such that $L(S_k/G_k) = \overline{P_k(K)}$. The supervisor S_k is used in the remaining part of the definition.
2) The language $P_{1\cup k}(K) \subseteq (E_1 \cup E_k)^*$ is controllable with respect to $L(G_1\|(S_k/G_k))$ and $E_{1+k,u} = E_u \cap (E_1 \cup E_k)$; equivalently,
$$\overline{P_{1\cup k}(K)}E_{1+k,u} \cap L(G_1\|(S_k/G_k)) \subseteq \overline{P_{1\cup k}(K)}$$
3) The language $P_{2\cup k}(K) \subseteq (E_2 \cup E_k)^*$ is controllable with respect to $L(G_2\|(S_k/G_k))$ and $E_{2+k,u} = E_u \cap (E_2 \cup E_k)$; equivalently,
$$\overline{P_{2\cup k}(K)}E_{2+k,u} \cap L(G_2\|(S_k/G_k)) \subseteq \overline{P_{2\cup k}(K)}$$

The conditions of Definition 7.2 can be checked by algorithms as is directly clear from the computational complexity of controllability in the case of only one subsystem. The computational complexity of checking conditional controllability is much less than that of the global system, $L(G_1)\|L(G_2)\|L(G_k)$. This is because instead of checking the controllability with global specification and system we check it only on the corresponding projections to $E_k \cup E_1$ and $E_k \cup E_2$. The projections are smaller when they satisfy the observer property.

Theorem 7.3: Consider Problem 7.1 of control for safety. There exists a set of supervisors (S_k, S_1, S_2) such that

$$L(S_1/[G_1\|(S_k/G_C)])\|L(S_2/[G_2\|(S_k/G_C)])\|L(S_k/G_k)) = \overline{K}, \quad (18)$$

if the specification language K is conditionally controllable with respect to (G_1, G_2, G_k) and $(E_{1,u}, E_{2,u}, E_{k,u})$.

At the time of submission of this paper, there is no result yet on the necessity condition of the safety.

The interest in Theorem 7.3 is in the computational saving of the computation of the supervisor, the distributed way of constructing successively the supervisors S_k, S_1, and S_2 is much less complex than that of the global supervisor constructed for the system $G_1 \| G_2 \| G_k$.

VIII. CONCLUDING REMARKS

The concepts of conditional independence of a tuple of generators given a third generator and that of a coordinator for modular discrete-event system have been introduced. It was established that using a coordinator for safety and for nonblockingness in a modular system the composite global supervisor satisfies the safety and the nonblockingness conditions. A construction of the overall supervisor for safety was proposed. More work on coordination control is needed: construction of minimal coordinators should be investigated.

REFERENCES

[1] C. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Boston: Kluwer Academic Publishers, 1999.

[2] B. Gaudin and H. Marchand, "Supervisory control of concurrent discrete event systems," IRISA, Rennes, Report Publication interne No. 1593, 2004.

[3] L. Feng, "Computationally efficient supervisory control design for discrete-event systems," Ph.D. dissertation, University of Toronto, Toronto, 2007.

[4] L. Feng and W. Wonham, "Computationally efficient supervisor design: Abstraction and modularity," in *Proc. 8th International Workshop on Discrete Event Systems*, S. Lafortune and F. Lin, Eds., IEEE. New York: IEEE, 2006, pp. 3–8.

[5] J. Komenda and J. H. van Schuppen, "Supremal sublanguages of general specification languages arising in modular control of discrete-event systems," in *Proc. 44th IEEE Conference on Decision and Control*. New York: IEEE Press, 2005, pp. 2775–2780.

[6] J. Komenda, J. H. van Schuppen, B. Gaudin, and H. Marchand, "Modular supervisory control with general indecomposable specification languages," in *Proc. 44th IEEE Conference on Decision and Control*. New York: IEEE Press, 2005, pp. 3474–3479.

[7] J. Komenda and J. H. van Schuppen, "Conditions structurelles dans le contrôle modulaire des systèmes à événements discrets concurrents," in *Proceedings Modélisation des Systèmes Ractifs (MSR) 2007*, E. Niel and J.-M. Miller, Eds., Ecole Normale Superior de Lyon. Paris: Hermès (Lavoisier), 2007, pp. 53–70.

[8] J. Komenda and J. H. van Schuppen, "Control of discrete-event systems with modular or distributed structure," *Theoretical Computer Science*, vol. 388, pp. 199–226, 2007.

[9] C. van Putten and J. van Schuppen, "Invariance properties of the conditional independence relation," *Ann. Probab.*, vol. 13, pp. 934–945, 1985.

[10] P. Ramadge and W. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, pp. 206–230, 1987.

[11] K.Schmidt, B. Gaudin, and H. Marchand, "Modular and decentralized supervisory control of concurrent discrete event systems using reduced system models," in *Proc. Int. Workshop on Discrete-Event Systems (WODES.2006)*. New York: IEEE, 2006, pp. 149–154.

[12] K. Schmidt, "Hierarchical control of decentralized discrete event systems," Ph.D. dissertation, Universiät Erlangen-Nürnberg, Erlangen, 2005.

[13] K. Wong, "On the complexity of projections of discrete-event systems," in *Proceedings International Workshop on Discrete-Event Systems (WODES'98)*. London: IEE, 1998, pp. 201–206.

[14] K. C. Wong and W. M. Wonham, "Modular control and coordination of discrete-event systems," *Discrete Event Dynamics Systems*, vol. 8, pp. 247–297, 1998.

[15] W. Wonham, *Lecture notes on control of discrete-event systems*. Toronto: University of Toronto, Department ECE, 2005.

APPENDIX

In this appendix proofs of Theorem 4.2 and Lemma 4.4 are stated.

Proof: of Theorem 4.2 (a) is a property known from concurrency theory. Let L_1, L_2 be conditionally independent given L_k, i.e. $E_r(L_1\|L_2) \cap E_1 \cap E_2 \subseteq E_k$. It is easy to show that the triple (L_1, L_2, L_c) is conditionally shuffle closed. We sketch the proof for $k = 1$. Let $t = s_1 t_{c1} \in L_1\|L_2\|L_k$ with $s_1 \in [(E_{1\setminus c}) \cup (E_{2\setminus c}]^*$ and $t_{c1} \in E_k^*$. Then $P_1(t) = P_1(s_1)P_1(t_{c1}) = P_{1\setminus c}(t)P_1(t_{c1}) \in L_1$, $P_1(t) = P_1(s_1)P_1(t_{c1}) = P_{1\setminus c}(t)P_1(t_{c1}) \in L_1$, and $P_k(t) = t_{c1} \in L_k$. For any element $\bar{t} \in \text{shuffle}(P_{1\setminus c}(s_1), P_{2\setminus c}(s_1))t_{c1} = P_{1\setminus c}^{-1}P_{1\setminus c}(s_1) \cap P_{2\setminus c}^{-1}P_{2\setminus c}(s_1)t_{c1}$ we get $P_1(\bar{t}) = P_{1\setminus c}(s_1)P_1(t_{c1}) \in L_1$. Similarly, $P_2(\bar{t}) \in L_2$, and finally $P_k(\bar{t}) = t_(c1) \in L_k$, hence $\bar{t} \in L_1\|L_2\|L_k$. The argument can be extended by induction along the string $t \in L_1\|L_2\|L_k$.
(b) $(b1) \Rightarrow (b2)$ Let the triple (L_1, L_2, L_k) is conditionally shuffle closed and let $t, \bar{t} \in E^*$ be such that $P_k(\bar{t}) = P_k(t), P_{1\cup k}(\bar{t}) = P_{1\cup k}(t)$, $P_{2\cup k}(\bar{t}) = P_{2\cup k}(t)$, and $\bar{t} \in L_1\|L_2\|L_k$. Let $\bar{t}$ have the following decomposition:

$$\bar{t} = s_1 t_{c_1} s_2 \dots s_k t_{c,k} s_{k+1}$$

with $t_{c_i} \in E_k^*$, $i = 1, \dots, k$ and $s_i \in [(E_1 \setminus E_k) \cup (E_2 \setminus E_k)]^*$. Then $P_k(t) = P_k(\bar{t}) = t_{c_1} \dots t_{c,k}$. Also, $P_{1\cup k}(t) = P_{1\cup k}(\bar{t}) = P_{1\setminus c}(s_1)t_{c_1} \dots t_{c,k}P_{1\setminus c}(s_{k+1})$ and $P_{2\cup k}(t) = P_{2\cup k}(\bar{t}) = P_{2\setminus c}(s_1)t_{c_1} \dots t_{c,k}P_{2\setminus c}(s_{k+1})$. It follows from the assumptions that $t \in \text{shuffle}(P_{1\setminus c}(s_1), P_{2\setminus c}(s_1))t_{c_1} \dots t_{c,k}\text{shuffle}(P_{1\setminus c}(s_{k+1})P_{2\setminus c}s_{k+1})$, i.e. $t \in L_1\|L_2\|L_k$ using conditional independence.
$(b2) \Leftrightarrow (b3)$ is easy: (b2) can be directly rewritten as $P_{1\cup k}(L_1\|L_2\|L_k)\|P_{2\cup k}(L_1\|L_2\|L_k)\|P_k(L_1\|L_2\|L_k) \subseteq L_1\|L_2\|L_k$, which is the nontrivial inclusion of the definition of conditional decomposability (b3).
$(b3) \Rightarrow (b1)$ Let $t, \bar{t} \in E^*$ be such that

$$\bar{t} = s_1 t_{c_1} s_2 \dots s_k t_{c,k} s_{k+1} \in L_1\|L_2\|L_k$$

with $t_{c_i} \in E_k^*$, $i = 1, \dots, k$ and $s_i \in [(E_1 \setminus E_k) \cup (E_2 \setminus E_k)]^*$ $i = 1, \dots, k+1$, and let $t \in \text{shuffle}(P_{1\setminus c}(s_1), P_{2\setminus c}(s_1))t_{c_1} \dots t_{c,k}\text{shuffle}(P_{1\setminus c}(s_{k+1})P_{2\setminus c}s_{k+1})$. Then $P_k(\bar{t}) = P_k(t) = t_{c_1} \dots t_{c,k}$, $P_{1\cup k}(\bar{t}) = P_{1\cup k}(t)$, and $P_{2\cup k}(\bar{t}) = P_{2\cup k}(t)$, and $\bar{t} \in L_1\|L_2\|L_k$. Then $t \in (P_{1\cup k})^{-1}P_{1\cup k}(L_1\|L_2\|L_k) \cap (P_{2\cup k})^{-1}P_{2\cup k}(L_1\|L_2\|L_k) \cap (P_k)^{-1}P_k(L_1\|L_2\|L_k) = P_{1\cup k}(L_1\|L_2\|L_k)\|P_{2\cup k}(L_1\|L_2\|L_k)\|P_k(L_1\|L_2\|L_k) = L_1\|L_2\|L_k$, hence $t \in L_1\|L_2\|L_k$ and (b1) holds. □

Proof: of Lemma 4.4 (i) It follows easily from Lemma 4.3 with $E_1 := E_k$ and $E_2 := E_1 \cup E_2$: $P_k(L_1\|L_2\|L_k) = P_k(L_1\|L_2)\|P_k(L_k) = P_k(L_1\|L_2) \cap L_k \subseteq L_k$, because both $P_k(L_k) = L_k$ and $P_k(L_1\|L_2)$ are languages over whole E_k.
(ii) Lemma 4.3 yields $P_{1\cup k}(L_1\|L_2\|L_k) = P_{1\cup k}((L_1\|L_k)\|L_2) = L_1\|L_k\|P^2_{2\cap[1\cup k]}(L_2) = L_1\|L_k \cap (P^{1\cup k}_{2\cap[1\cup k]})^{-1}P^2_{2\cap[1\cup k]}(L_2) \subseteq L_1\|L_k$, where $P^2_{2\cap[1\cup k]} : E_2^* \to (E_2 \cap E_{1\cup k})^*$ and $(P^{1\cup k}_{2\cap[1\cup k]})^{-1} : \text{Pwr}(E_2 \cap E_{1\cup k})^* \to \text{Pwr}E^*_{1\cup k}$.
(iii) proof is similar to (ii) □

Acknowledgment

Partial financial support of Grant GA AV No. B100190609 and the Academy of Sciences of the Czech Republic, Institutional Research Plan No. AV0Z10190503. is gratefully acknowledged.