A Bridge between Decentralized and Coordination Control

Jan Komenda and Tomáš Masopust

***Abstract*— In decentralized supervisory control, several local control agents (supervisors) cooperate to achieve a common goal, expressed by a safety specification and/or by nonblockingness. It is well-known that coobservability is the key condition to achieve the specification as the resulting language of the controlled system. One of the most important problems is to compute a coobservable sublanguage of the specification. This paper shows how recent results in coordination control of modular discrete-event systems help to construct a coobservable sublanguage in a computationally cheap way. The impact of coordination control on decentralized control is discussed in detail.**

I. Introduction

Decentralized supervisory control of discrete-event systems in the Ramadge-Wonham framework has been developed in [19]. It is based on distributing overall actuator and sensor capabilities among several local control agents, called supervisors, and motivated by the decrease of computational complexity—the overall task is divided into local tasks for individual supervisors, each of which reacts according to a partial observation of the system's moves. Each supervisor can issue its own control decision on enabling or disabling an event based on its observation of the system behavior, which is formally given as a projection of the system behavior. The global control action of the decentralized control architecture is then given by a fusion rule on the local control actions. There are many different control policies that are based on two elementary ones, namely the *conjunctive and permissive* (C & P) policy, and the *disjunctive and antipermissive* (D & A) policy. For any such a decentralized control architecture, a corresponding notion of *coobservability* has been proposed, which together with *controllability* form the necessary and sufficient conditions to achieve a specification as the resulting behavior of the closed-loop system.

However, almost all results available in the literature provide existential results. There are only a few papers providing constructive results, namely, how to compute a controllable and coobservable sublanguage of a specification that fails to satisfy these properties. It is in general considered as a computationally difficult problem, but the existence of a set of local supervisors that enforce the safety specification (the behavior of the controlled system is included in the specification) is still decidable when nonblockingness is not required unlike the general problem. Indeed, if the marked language of the controlled system has to be included in the specification so that the controlled system is nonblocking, then the existence of such local supervisors has been shown undecidable in [21], [22].

Another approach to ensure coobservability of a specification is to extend locally observable events by communication among local supervisors. There exist decentralized control problems that cannot be solved without enriching locally observable events via communication. Decentralized control with communicating supervisors, where an occurrence of transitions visible to one supervisor can be communicated to other supervisors, has been studied in [1], [17]. Nowadays, there exist more advanced architectures of decentralized supervisory control, such as with conditional decision (inferencing) [25] or even multi-level inferencing [14], [20]. A general approach consisting in several decentralized supervisory control architectures running in parallel has been proposed in [4]. Note that only the original C & P architecture is considered in this paper, because it is closely related to the coordination control architecture of [9], [12]. The other decentralized control architectures would require a different coordination control architecture of modular systems that should be developed first so that it matches (can be applied to) these architectures.

In this paper, we focus on the computation of a controllable and coobservable sublanguage using the coordination control approach [9], [12], which can be seen as a modular control with communication, where local supervisors communicate the occurrence of some events (called coordinator events) via a coordinator.

Our study is limited to the original (C & P) control architecture [19], [24], and is motivated by the relationship between decentralized and more structured modular supervisory control and their key concepts: (C & P) coobservability and decomposability. This relationship has been investigated in [8], where the decentralized supervisory control framework is plugged into the supervisory control framework, which makes it possible to benefit from richer constructive results available in modular supervisory control. The approach is based on the concurrent (decomposable) overapproximation of the decentralized control plant, where no concurrent structure is known. In decentralized supervisory control, there is no assumption on the plant. In [8], both the system and the specification are replaced with their infimal decomposable superlanguages with respect to local observation alphabets. However, in the (likely) case where the decentralized control specification fails to be decomposable, one only computes a solution of the decentralized control problem for this new specification, and the controllable and coobservable sublanguage computed using modular control often fails to be included in the specification. Otherwise

Institute of Mathematics, Academy of Sciences of the Czech Republic, Žižkova 22, 616 62 Brno, Czech Republic. Emails: komenda@ipm.cz, masopust@math.cas.cz

stated, the authors of [8] assume that the constructed sublanguage is included in the specification.

This is the point, where our coordination control comes into the picture because we can benefit from the recent constructive results. Coordination control overcomes the problem of an indecomposable specification by making it decomposable using a coordinator (we then speak about *conditional decomposability*). Moreover, the approach of [8] requires *mutual controllability* among projected systems to ensure coobservability of the decomposable over-approximation of the specification. This can be omitted if constructive results of coordination control are applied.

Decomposable over-approximations have also been considered in [7]. In that paper, however, decomposability of the specification is an additional assumption, whereas in this paper, coobservability is enforced by the construction as a consequence of the coordination control theory [12].

In this paper, we summarize how decentralized supervisory control can benefit from constructive results of modular and coordination control. A procedure to compute a controllable and coobservable sublanguage of a specification with respect to possibly extended local observations (enriched by communication) is presented. The paper is organized as follows. Section II recalls the Ramadge-Wonham supervisory control framework, Section III presents the decentralized supervisory control problem, and Section IV gives a brief summary of constructive results of coordination control. Section V shows how decentralized supervisory control can benefit from recent results of coordination control using the decomposable over-approximation of the plant. Concluding remarks together with hints on future extensions of the proposed approach can be found in Section VI.

II. Control of Discrete-Event Systems

Before the technical development, basic notations of supervisory control needed in this paper are recalled, see [3], [23]. An alphabet, A, is a finite nonempty set. The set of all finite words over A is denoted by A^*. The empty string is denoted by ε. A language over A is a subset of A^*.

A *generator* is a quintuple

$$G = (Q, A, f, q_0, Q_m)\,,$$

where Q is a finite set of *states*, A is an alphabet (of *events*), $f : Q \times A \to Q$ is a *partial transition function*, $q_0 \in Q$ is the *initial state*, and $Q_m \subseteq Q$ is a set of *marked states*. As usual, f is extended to $f : Q \times A^* \to Q$. The *language generated* by G is defined as the set $L(G) = \{s \in A^* \mid f(q_0, s) \in Q\}$, and the *marked language* of G as the set $L_m(G) = \{s \in A^* \mid f(q_0, s) \in Q_m\}$.

For alphabets $A_0 \subseteq A$, a *projection* $P : A^* \to A_0^*$ is a morphism defined by

$$P(a) = \begin{cases} \varepsilon, & \text{if } a \in A \setminus A_0 \\ a, & \text{if } a \in A_0\,. \end{cases}$$

The *inverse image* of P is defined as

$$P^{-1}(a) = \{s \in A^* \mid P(s) = a\}\,.$$

These definitions can naturally be extended to languages. We use the notation $A_{i+j} = A_i \cup A_j$ to denote the union of the corresponding alphabets, and

$$P^{i+k}_j : A^*_{i+k} \to A^*_j$$

to denote the projection from A^*_{i+k} to A^*_j. Similarly, notation $P_{i+k} : A^* \to A^*_{i+k}$ stands for the projection from A^* to A^*_{i+k}.

A *synchronous product* of languages $L_i \subseteq A_i^*$, for $i = 1, 2, \ldots, n$, is defined as

$$\Big\|_{i=1}^{n} L_i = \bigcap_{i=1}^{n} P_i^{-1}(L_i) \subseteq A^* = (\bigcup_{i=1}^{n} A_i)^*\,,$$

where $P_i : A^* \to A_i^*$ is a projection. For two generators G_1 and G_2, the synchronous product of generators is defined in [3], which satisfies $L(G_1 \| G_2) = L(G_1) \| L(G_2)$ and $L_m(G_1 \| G_2) = L_m(G_1) \| L_m(G_2)$.

The *prefix closure* $\overline{L}$ of a language L is the set of all prefixes of all its words; L is prefix-closed if $L = \overline{L}$.

Let L be a prefix-closed language over A, and let $A_u \subseteq A$ be the set of uncontrollable events. A language $K \subseteq L$ is *controllable* with respect to L and A_u if

$$\overline{K} A_u \cap L \subseteq \overline{K}\,.$$

For uncontrollable languages, controllable sublanguages are considered. The notation $\sup \mathrm{C}(K, L, A_u)$ denotes the supremal controllable sublanguage of K with respect to L and A_u, which always exists and equals to the union of all controllable sublanguages of K, see [3].

The distributed control synthesis of a modular discrete-event system is a procedure where the control synthesis is carried out separately for each local supervisor. The global supervisor then formally consists of the synchronous product of local supervisors, although it is not computed in practice. In terms of behaviors, the optimal global control synthesis is represented by the closed-loop language

$$\sup \mathrm{C}(K, L, A_u) = \sup \mathrm{C}(\|_{i=1}^{n} K_i, \|_{i=1}^{n} L_i, A_u)\,.$$

In the decentralized control synthesis, the specification K is replaced by $K_i = K \cap P_i^{-1}(L_i)$ and the synthesis is done as for local specifications or by using the notion of partial controllability [6]. Notice the difference with decentralized control of monolithic plants studied in [24], where several control agents have different observations, but the system has no modular structure consisting of subsystems running in parallel. The purely modular control synthesis is not always possible, which motivated the study of coordination control.

Finally, we recall two important properties used in our approach. Let $P : A^* \to A_k^*$, where $A_k \subseteq A$, be a projection. This projection will later correspond to abstraction of the plant on the high-level coordinator alphabet. Since coordination control combines decentralized and hierarchical control, we make use of two major properties of hierarchical control stated below.

(i) Projection P is an *L-observer* for a language $L \subseteq A^*$ if for all $s \in \overline{L}$, if $P(s)t \in P(L)$, then there exists $u \in A^*$ such that $su \in L$ and $P(u) = t$, see Fig. 1.

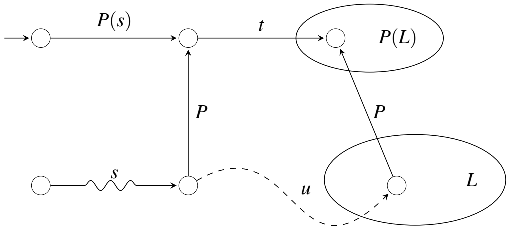

Fig. 1. Illustration of the observer property.

Observer property states that if a projection $P(s)$ (abstraction to the high-level) of a string $s \in \overline{L}$ belongs to the prefix-closure of the projected language then for any extension $P(s)t$ to the projected language, there must exist an extension u of the string s to language L itself that is projection compatible, i.e., $P(su) = P(s)t$ (meaning also $P(u) = t$).

(ii) Projection P is *output control consistent* (OCC) for a language $L \subseteq A^*$ if for every $s \in \overline{L}$ of the form $s = \sigma_1\sigma_2 \ldots \sigma_\ell$ or $s = s'\sigma_0\sigma_1 \ldots \sigma_\ell$, for $\ell \geq 1$, where $\sigma_0, \sigma_\ell \in A_k$ and $\sigma_i \in A \setminus A_k$, for $i = 1, 2, \ldots, \ell - 1$, if $\sigma_\ell \in A_u$, then $\sigma_i \in A_u$, for all $i = 1, 2, \ldots, \ell - 1$.

OCC property states that there is enough controllable events on the high level (from A_k) in the sense that if we need to forbid a high level uncontrollable event then the nearest upstream controllable events must be a high level event. Otherwise stated, all upstream low level events must be uncontrollable and therefore even if these events would be in the high level it would not change anything as no supervisor can disable them. Indeed, it follows from the intuition that if one of these low level events is controllable then on the high level we loose the control power on this event and hence we need to disable a high-level upstream controllable event, which may obviously lead to a loss of optimality of the hierarchical supervisory control with respect to the low level supervisor.

If L is represented by a generator with n states, then the complexity to verify the properties is $O(n^2)$ as shown in [2], [5].

III. Decentralized Supervisory Control: The Synthesis Problem

In decentralized supervisory control, two or more local supervisors receive different partial observations of the system. Since communication of all local observations is either not possible or too costly, the partial observations of the local supervisors differ. Decentralized supervisory control consists in considering local supervisors $(S_i)_{i=1}^n$ and distributing the alphabet of controllable and observable events into locally controllable and locally observable events, denoted by $(A_{c,i})_{i=1}^n$ and $(A_{o,i})_{i=1}^n$, respectively. Projections to locally observable events are denoted by $P_i : A^* \to A_{o,i}^*$. The action of P_i is to delete events that are not observable by supervisor S_i. Furthermore, we denote $A_c = \cup_{i=1}^n A_{c,i}$, $A_o = \cup_{i=1}^n A_{o,i}$, $A_u = A \setminus A_c$, and $A_{uo} = A \setminus A_o$.

Let G be a generator. A local supervisor S_i is represented by a mapping $S_i : P_i(L(G)) \to \Gamma_i$, where $\Gamma_i = \{\gamma \subseteq A : \gamma \supseteq (A \setminus A_{c,i})\}$ is the set of local control patterns, and $S_i(s)$ represents the set of locally enabled events when S_i observes a string $s \in A_{o,i}^*$. The associated control law of the local supervisor S_i is $S_i(s) = (A \setminus A_{c,i}) \cup \{a \in A_{c,i}$: there exists $s' \in K$ with $P_i(s') = P_i(s)$ and $s'a \in K\}$. The global control law S is then the conjunction of local supervisors S_i given by $S(w) = \cap_{i=1}^n S_i(P_i(w))$, where $w \in A^*$.

The necessary and sufficient conditions for a given specification K to be achieved by a joint action of local supervisors are controllability and coobservability. The definition of coobservability from [19] can be extended from two to $n \geq 2$ supervisors.

Definition 1 (C & P coobservability): Let L be a prefix-closed language. A language $K \subseteq L$ is *C & P coobservable* with respect to L and $(A_{o,i})_{i=1}^n$ if for all $s \in \overline{K}$, $a \in A_c$, and $sa \in L \setminus \overline{K}$, there exists $i \in \{1, 2, \ldots, n\}$ such that $a \in A_{c,i}$ and $(P_i^{-1}(P_i(s))\{a\} \cap \overline{K} = \emptyset$. □

The intuition behind C & P coobservability is as follows. If after a given string s from the specification the continuation by an event a is illegal, i.e., it does not exist in the specification while remaining within the plant language, then there must exist at least one local supervisor that has control power on this event ($a \in A_{c,i}$) that can issue the decision "disable the event a" without ambiguity (all observation equivalent strings are illegal as well). This is because in the conjunctive architecture an event is enabled if and only if all local supervisors enable this event, hence the local disabling decision must be made unambiguously. Coobservability, which is decidable in polynomial time [18] in the number of states of the plant and the specification (but in exponential time in the number of local supervisors!), is needed for the existence of local supervisors that jointly achieve the specification. In some cases where the structure of the intersection between locally observable alphabets is simple (e.g., it is empty, or the intersection of all pairs of locally observable alphabets is always the same), coobservability is strongly related to conditional decomposability defined below and, thus, can be decided in polynomial time in the number of local agents [11].

The control law of local supervisors associated to the C & P architecture is called permissive, since the default action is to enable an event whenever a local supervisor has an ambiguity what to do with it. With the permissive local policy we always achieve all strings in the specification. The only concern is then *safety*, expressed by C & P coobservability, which states that there always exists a local supervisor that is sure to disable an event resulting in an illegal string, which is the motivation for Definition 1. As shown in [19], coobservability with controllability are the necessary and sufficient conditions to achieve the specification as the resulting closed-loop language, whence the interest in the computation of controllable and coobservable sublanguages.

There is a natural counterpart of the C & P control architecture, called D & A (disjunctive and antipermissive), with a corresponding notion of coobservability, but this is not studied in this paper. Thus, we call a C & P coobservable sublanguage simply coobservable.

A conceptually simpler condition than coobservability is known as decomposability.

Definition 2: A language K is *decomposable* with respect to alphabets $(A_i)_{i=1}^n$ and L if $K = \|_{i=1}^n P_i(K) \cap L$. □

Note that the inclusion $K \subseteq \|_{i=1}^n P_i(K) \cap L$ holds true whenever $K \subseteq L$. Intuitively, K is decomposable with respect to $(A_i)_{i=1}^n$ and L whenever one can infer $s \in K$ from $P_i(s) \in P_i(K)$ and $s \in L$. Otherwise stated, the language K can be recovered from its projections $P_i(K)$ and L.

We recall from Proposition 4.3 in [19] that under mild (and reasonable) assumptions on the structure of locally controllable and locally observable events, decomposability implies coobservability. Since the result in [19] is only for two control agents, and it turns out that for this implication a weaker condition is needed than for the equivalence of both properties studied in [19], we state the following auxiliary result.

Proposition 3: Assume that K is decomposable with respect to $(A_{o,i})_{i=1}^n$ and L, and that for $i = 1, 2, \ldots, n$, $A_{o,i} \cap A_c \subseteq A_{c,i}$. Then K is coobservable with respect to L and $(A_{o,i})_{i=1}^n$.

Proof: We prove it by contradiction. Assume that K is decomposable with respect to $(A_{o,i})_{i=1}^n$ and L, but not coobservable with respect to L and $(A_{o,i})_{i=1}^n$. Let $s \in \overline{K}$, $a \in A_c$, and $sa \in L \setminus \overline{K}$ as in Definition 1. Then, by the assumption, for each $i \in \{1, 2, \ldots, n\}$ for which $a \in A_{c,i}$, there exists $s_i \in \overline{K}$ such that $s_i a \in \overline{K}$ and $P_i(s_i) = P_i(s)$. That is, $P_i(sa) = P_i(s_i a) \in P_i(\overline{K})$, hence $sa \in P_i^{-1} P_i(\overline{K})$. Similarly for each $i \in \{1, 2, \ldots, n\}$ for which $a \notin A_{c,i}$, we have that $a \notin A_{o,i} \cap A_c$. However, $a \in A_c$, hence $a \notin A_{o,i}$, which means that $P_i(a) = \varepsilon$. This means that $P_i(sa) = P_i(s) \in P_i(\overline{K})$, that is, $sa \in P_i^{-1} P_i(\overline{K})$. Since $sa \in L$, we have altogether that $sa \in \cap_{i=1}^n P_i^{-1} P_i(\overline{K}) \cap L$. Then decomposability of K with respect to $(A_{o,i})_{i=1}^n$ and L implies that $sa \in \overline{K}$ as well, which is a contradiction. ■

Lemma 4: The property $A_{o,i} \cap A_c \subseteq A_{c,i}$, for $i = 1, 2, \ldots, n$, is equivalent to $A_{o,i} \cap A_{c,j} \subseteq A_{c,i}$, for $i, j = 1, 2, \ldots, n$.

Proof: Notice that $A_{o,i} \cap A_c = A_{o,i} \cap (\cup_{j=1}^n A_{c,j}) = \cup_{j=1}^n (A_{o,i} \cap A_{c,j})$. Now, $\cup_{j=1}^n (A_{o,i} \cap A_{c,j}) \subseteq A_{c,i}$ if and only if $A_{o,i} \cap A_{c,j} \subseteq A_{c,i}$, for all $i, j = 1, 2, \ldots, n$. ■

Finally, we need the following special instance of decomposability for $L = A^*$, which is known in the literature as *separability* [6].

Definition 5: A language K is *separable* with respect to alphabets $(A_i)_{i=1}^n$ if $K = \|_{i=1}^n P_i(K)$. □

The following simple lemma states that language K is separable if and only if it is given as a parallel composition of n languages (over the required alphabets).

Lemma 6: A language $K \subseteq (A_1 \cup A_2 \cup \ldots \cup A_n)^*$ is separable with respect to alphabets A_1, A_2,..., A_n, if and only if there exist languages $M_i \subseteq A_i^*$, $i = 1, 2, \ldots, n$, such that $K = \|_{i=1}^n M_i$.

Proof: If $K = \|_{i=1}^n P_i(K)$, define $M_i = P_i(K)$, for $i = 1, 2, \ldots, n$.

On the other hand, assume that there exist languages $M_i \subseteq A_i^*$, $i = 1, 2, \ldots, n$, such that $K = \|_{i=1}^n M_i$. Obviously, $P_i(K) \subseteq M_i$, $i = 1, 2, \ldots, n$, which implies that $\|_{i=1}^n P_i(K) \subseteq K$. As it always holds that $K \subseteq P_i^{-1}[P_i(K)]$, definition of the synchronous product implies that $K \subseteq \|_{i=1}^n P_i(K)$. ■

Separability is equivalent to the inclusion $\|_{i=1}^n P_i(K) \subseteq K$, while decomposability with respect to L is equivalent to $\|_{i=1}^n P_i(K) \cap L \subseteq K$ only if $K \subseteq L$. This is, however, not such a big issue, because in the case $K \nsubseteq L$ the refined specification $K \cap L$ is used. More precisely, the following statement holds true.

Proposition 7 ([8]): If K is separable, then $K \cap L$ is decomposable with respect to $(A_{o,i})_{i=1}^n$ and L, that is, $K \cap L = \|_{i=1}^n P_i(K \cap L) \cap L$. ■

The following theorem follows from Propositions 3 and 7.

Theorem 8: Assume that $A_{o,i} \cap A_c \subseteq A_{c,i}$, for $i = 1, 2, \ldots, n$. If K is separable with respect to $(A_{o,i})_{i=1}^n$, then $K \cap L$ is coobservable with respect to $(A_{o,i})_{i=1}^n$ and L. ■

IV. Coordination Control Synthesis: Constructive Results

Recall that the main constructive result of coordination control enables in particular the computation of the supremal controllable sublanguage in a distributed way. It relies on the concept of conditional decomposability.

To later distinguish between decentralized and coordination control approaches, we use A to denote alphabets in decentralized control and E to denote alphabets in coordination control. Let us recall that the main distinguishing feature of our coordination control approach is that the plant generator over the alphabet E has an explicit modular (concurrent) structure in the form of synchronous product of local automata over the alphabets $(E_i)_{i=1}^n$ with $E = \cup_{i=1}^n E_i$. Coordination control is then a generalization of the purely modular synthesis. It is based on the concept of conditional decomposability that generalizes language separability by using the so-called coordinator alphabet E_k.

Definition 9: A language K is *conditionally decomposable* with respect to alphabets $(E_i)_{i=1}^n$ and E_k, where $\cup_{i \neq j}(E_i \cap E_j) \subseteq E_k$ if

$$K = \|_{i=1}^n P_{i+k}(K),$$

where $P_{i+k} : (\cup_{i=1}^n E_i)^* \to E_{i+k}^*$, for $E_{i+k} = E_i \cup E_k$. □

Recall that there always exists E_k that makes language K conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k [11]. Moreover, K is conditionally decomposable if and only if there exist $M_i \subseteq E_{i+k}^*$ such that $K = \|_{i=1}^n M_i$. In that case, $P_{i+k}(K) \subseteq M_i$, which means that even though several tuples of (local) languages M_i may exist, $(P_{i+k}(K))_{i=1}^n$ form the smallest decomposition. Actually, $\|_{i=1}^n P_{i+k}(K)$ is the infimal superlanguage of K that is conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k.

Let $K \subseteq L = \|_{i=1}^n L_i$ be languages over $\cup_{i=1}^n E_i$, where $L_i \subseteq E_i^*$. Assume that K is conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k, and define the languages

$$\begin{aligned} L_k &= \|_{i=1}^n P_k(L_i), \\ \sup \mathrm{C}_k &= \sup \mathrm{C}(P_k(K), L_k, E_{k,u}), \\ \sup \mathrm{C}_{i+k} &= \sup \mathrm{C}(P_{i+k}(K), L_i \| \sup \mathrm{C}_k, E_{i+k,u}). \end{aligned} \tag{1}$$

Below the main result of [12] is recalled.

Theorem 10: Consider languages as defined in (1) with K and L being prefix-closed. Let the projection P_k^{i+k} be an $(P_i^{i+k})^{-1}(L_i)$-observer and OCC for $(P_i^{i+k})^{-1}(L_i)$, for $i = 1, 2, \ldots, n$. Then, $\|_{i=1}^n \sup \mathrm{C}_{i+k}$ is controllable with respect to L and E_u. ∎

It is equal to the supremal controllable sublanguage in some cases provided in [12], but it does not play a role in this paper because it is known that supremal coobservable sublanguages do not exist anyway.

Recently, we have shown in [10], [13] the following. Note that K need not be prefix-closed in this case.

Theorem 11: Consider languages as defined in (1) with L prefix-closed. If $\sup \mathrm{C}_k \subseteq P_k(\sup \mathrm{C}_{i+k})$, for $i = 1, 2, \ldots, n$, then $\|_{i=1}^n \sup \mathrm{C}_{i+k}$ is controllable with respect to L and E_u. ∎

V. Application of Coordination and Modular Control to Decentralized Control

Return to the setting of decentralized control, that is, given sets of local observable events $(A_{o,i})_{i=1}^n$ and local controllable events $(A_{c,i})_{i=1}^n$. Unlike [8] we do not need to assume that $A_c \subseteq A_o$. This was not needed in Proposition 3, because this inclusion is only needed for the converse implication that we do not need in this paper. Another difference with [8] is that we do not use the additional module based on projection on unobservable events, where this condition has been used.

We now present three different possibilities of computing the controllable and coobservable sublanguage based on the over-approximation of the plant language L by separable superlanguages and by making the specification K conditionally decomposable. To transform the decentralized problem to the coordination control problem, we set

$$E_i = A_{o,i} \quad \text{and} \quad E_{c,i} = A_{o,i} \cap A_{c,i}.$$

We over-approximate the plant language L by the new modular plant $\|_{i=1}^n P_i(L)$, that is, by the parallel composition of projections to events observable by local control agents. Next, we need to find an extension E_k of local alphabets so that K is conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k. Similarly as above, P_{i+k} denotes the projection from $E^* = (\cup_{i=1}^n E_i \cup A_{uo})^*$ to $E_{i+k} = (E_i \cup E_k)^*$, and P_i the projection from E^* to E_i^*.

In coordination control [12] it is assumed that shared events have the same controllability status in all components. We need to assume that $A_{o,i} \cap A_{c,j} \subseteq A_{c,i}$, for $i, j = 1, 2, \ldots, n$, for two different reasons.

First, due to Lemma 4, it is equivalent to $A_{o,i} \cap A_c \subseteq A_{c,i}$, for $i = 1, 2, \ldots, n$, which is a condition that ensures that separability implies coobservability (cf. Theorem 8). Let us point out that Theorem 8 is the core of our approach: the computation based on coordination control will naturally lead to a separable controllable sublanguage of the specification, hence coobservability of the original will automatically hold by our construction.

The second reason why we need the event set assumption from decentralized control is that it automatically ensures the shared-event status condition needed in modular and coordination control. More precisely, the following lemma holds true.

Lemma 12: If $A_{o,i} \cap A_{c,j} \subseteq A_{c,i}$, for $i, j = 1, 2, \ldots, n$, then $(E_i)_{i=1}^n$ and $(E_{c,i})_{i=1}^n$ defined above satisfy $E_i \cap E_{c,j} \subseteq E_{c,i}$, for all $i, j = 1, 2, \ldots, n$.

Proof: The assumption implies that for all $i, j = 1, 2, \ldots, n$, $E_i \cap E_{c,j} = A_{o,i} \cap (A_{o,j} \cap A_{c,j}) \subseteq A_{o,i} \cap A_{c,j} \subseteq A_{c,i}$ and, trivially, $E_i \cap E_{c,j} = A_{o,i} \cap (A_{o,j} \cap A_{c,j}) \subseteq A_{o,i}$, hence $E_i \cap E_{c,j} \subseteq A_{c,i} \cap A_{o,i} = E_{c,i}$, which was to be shown. ∎

A. Coordination Control with Observer and OCC

An important feature of our coordination control approach is that by computing the controllable sublanguage according to Theorem 10, we automatically obtain a decomposable (that is, coobservable) sublanguage. More precisely, we have the following result; here $E_{i+k,u} = E_{i+k} \cap A_u$.

Theorem 13: Let $K \subseteq L$ be prefix-closed languages, and let K be conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k. Let P_k^{i+k} be an $(P_i^{i+k})^{-1}(P_i(L))$-observer and OCC for $(P_i^{i+k})^{-1}(P_i(L))$, for $i = 1, 2, \ldots, n$. Then the language

$$M = \|_{i=1}^n \sup \mathrm{C}(P_{i+k}(K), P_i(L) \| \sup \mathrm{C}_k, E_{i+k,u})$$

is a sublanguage of K controllable with respect to L and E_u, and coobservable with respect to L and $(E_{i+k})_{i=1}^n$.

Proof: By Theorem 10, replacing L_i with $P_i(L)$ in (1), M is a controllable sublanguage of K with respect to $\|_{i=1}^n P_i(L)$ and E_u. Since $L \subseteq \|_{i=1}^n P_i(L)$, M is controllable with respect to L and E_u.

By the note below Definition 9, M is also conditionally decomposable, that is, separable with respect to $(E_{i+k})_{i=1}^n$. Since $E_i \cap E_{c,j} \subseteq E_{c,i}$, Theorem 8 shows that M is coobservable with respect to L and $(E_{i+k})_{i=1}^n$. ∎

Although decomposability, observer, and OCC conditions might seem strong, there always exists E_k such that these conditions are all satisfied. In some exceptional cases, it might be needed to take the whole alphabet $E_k = E$, which amounts to the need to communicate all events. This is not a surprise because there are small examples, where all events must be communicated between local supervisors to achieve the specification, but this is not a typical situation.

B. Coordination Control without Observer and OCC

In some cases, where observer and OCC properties are expensive to impose (in the sense that too many events must be communicated via the coordinator for the properties to hold) we can benefit from another result of coordination control.

Theorem 14: Let $K \subseteq L = \overline{L}$ be languages, and let K be conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k. If $\sup \mathrm{C}_k \subseteq P_k(\sup \mathrm{C}_{i+k})$, for $i = 1, 2, \ldots, n$, computed as in (1), then $\|_{i=1}^n \sup \mathrm{C}(P_{i+k}(K), P_i(L) \| \sup \mathrm{C}_k, E_{i+k,u})$ is a sublanguage of K controllable with respect to L and E_u, and coobservable with respect to L and $(E_{i+k})_{i=1}^n$.

Proof: If $\sup \mathrm{C}_k \subseteq P_k(\sup \mathrm{C}_{i+k})$, for $i = 1, 2, \ldots, n$, then $\|_{i=1}^n \sup \mathrm{C}(P_{i+k}(K), P_i(L) \| \sup \mathrm{C}_k, E_{i+k,u})$ is a controllable sublanguage of K by Theorem 11. The rest of the proof is the same as in the previous theorem. ∎

Hence, it might be more advantageous in certain cases to check if $\sup \mathrm{C}_k \subseteq P_k(\sup \mathrm{C}_{i+k})$ instead of checking observer and OCC conditions that are not necessary in Theorem 11.

C. Modular Control with Mutual Controllability

Finally, we present an approach which uses modular control and conditional decomposability of K. This approach has been partially presented in [8] in the case, where no information exchange is allowed. In this case, we over-approximate the plant language by $\|_{i=1}^n P_{i+k}(L)$, where E_k is computed so that $K = \|_{i=1}^n P_{i+k}(K)$, i.e., K is conditionally decomposable.

Les us recall the concept of mutual controllability of [16].

Definition 15: Prefix-closed languages $L_i \subseteq A_i^*$, where $i = 1, 2, \ldots, n$, are *mutually controllable* if for all $i, j = 1, 2, \ldots, n$, $L_j(A_{j,u} \cap A_i) \cap P_j(P_i)^{-1}(L_i) \subseteq L_j$. □

The following result on the compatibility between supremal controllable sublanguages and the synchronous composition operator has been first shown in [16].

Proposition 16: Assume that $A_{o,i} \cap A_{c,j} \subseteq A_{c,i}$. If the prefix-closed languages $L_i \subseteq A_i^*$, for $i = 1, 2, \ldots, n$, are mutually controllable, then for any decomposable specification $K \subseteq L$,

$$\|_{i=1}^n \sup \mathrm{C}(K_i, L_i, A_{i,u}) = \sup \mathrm{C}(\|_{i=1}^n K_i, \|_{i=1}^n L_i, A_u)$$

holds. ∎

Although it is restrictive to require that $P_i(L)$ and $P_j(L)$ are mutually controllable, it may well be that $P_{i+k}(L)$ and $P_{j+k}(L)$ are mutually controllable for a fairly small coordinator alphabet E_k. However, if we do not require K to be conditionally decomposable, we cannot guarantee that the resulting supremal controllable sublanguage is included in K, which is the main issue with the approach of [8]. Thus, we also require that K is conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k (note that E_k must be the same in both assumptions) to apply Proposition 16 in the context of a conditionally decomposable specification.

Proposition 17: Let $K \subseteq L$ be prefix-closed languages, and let K be conditionally decomposable with respect to $(E_i)_{i=1}^n$ and E_k. If $P_{i+k}(L)$ and $P_{j+k}(L)$ are mutually controllable, for $i, j = 1, 2, \ldots, n$, then the language

$$\|_{i=1}^n \sup \mathrm{C}(P_{i+k}(K), P_{i+k}(L), E_{i+k,u})$$

is a sublanguage of K controllable with respect to L and E_u, and coobservable with respect to L and $(E_{i+k})_{i=1}^n$.

Proof: Let M denote the language

$$\|_{i=1}^n \sup \mathrm{C}(P_{i+k}(K), P_{i+k}(L), E_{i+k,u}) \,.$$

Since $\sup \mathrm{C}(P_{i+k}(K), P_{i+k}(L), E_{i+k,u}) \subseteq P_{i+k}(K)$ we have that $M \subseteq \|_{i=1}^n P_{i+k}(K) \subseteq K$ due to conditional decomposability. Again, by Theorem 10, M is a controllable sublanguage of K with respect to $\|_{i=1}^n P_i(L)$ and E_u. Since $L \subseteq \|_{i=1}^n P_i(L)$, M is controllable with respect to L and E_u as well.

Let us notice that M is separable with respect to extended alphabets $(E_{i+k})_{i=1}^n$ (cf. Lemma 6) because it can be computed in a modular way under the assumption of mutual controllability of local plants. Hence, according to Theorem 8, $M \cap L = M$ is coobservable with respect to L and $(E_{i+k})_{i=1}^n$. Similarly as in the previous sections, it is not needed that L is conditionally decomposable with respect to the same alphabets as K, because $L \subseteq \|_{i=1}^n P_{i+k}(L)$ guarantees that controllability of M with respect to $\|_{i=1}^n P_{i+k}(L)$ implies controllability with respect to L. Altogether, M is a sublanguage of K controllable with respect to L and E_u, and coobservable with respect to L and $(E_{i+k})_{i=1}^n$. ∎

Note that mutual controllability holds true if all shared events are controllable, which can be written in our setting as $A_{o,i} \cap A_{o,j} \subseteq A_c$, but this is quite a strong assumption. Note that if $P_{i+k}(L)$ and $P_{j+k}(L)$ are not mutually controllable, we can use the technique of [15] to modify the plant so that the resulting plant is mutually controllable.

D. Example

Let $K = \overline{\{aa, ba, bbd, abc\}}$, $L = \overline{\{aac, abc, bac, bbd\}}$, $A_{o,1} = A_{c,1} = \{a, c\}$, and $A_{o,2} = A_{c,2} = \{b, d\}$. Then K is not coobservable with respect to L and $(A_{o,i})_{i=1}^2$, because none of the two supervisors is able to distinguish between the strings $s = ab$ and $s' = ba$, and the continuation of ba by c within the plant leads outside the specification while the continuation of ab by c remains within the specification.

We will proceed according to results in Section V-C, namely Proposition 17 will be applied. Thus, we set $E_i = A_{o,i}$, $i = 1, 2$. Note that $E_1 \cap E_2 = \emptyset$. We need to find $E_k \supseteq \emptyset$ such that K becomes conditionally decomposable with respect to projections to E_{1+k} and E_{2+k}. It is sufficient to take $E_k = \{b\}$, which says that each occurrence of b must be communicated between the two supervisors via a coordinator. Furthermore, the only shared event between E_{1+k} and E_{2+k}, namely b, is controllable. Hence, $P_{1+k}(L)$ and $P_{2+k}(L)$ are mutually controllable.

- $\sup \mathrm{C}(P_{1+k}(K), P_{1+k}(L), E_{1+k,u}) = \overline{\{aa, ba, bb, ab\}}$,
- $\sup \mathrm{C}(P_{2+k}(K), P_{2+k}(L), E_{2+k,u}) = \overline{\{bbd\}}$.

It is now easy to see that $K = \|_{i=1}^n \sup \mathrm{C}(P_{i+k}(K), P_{i+k}(L), E_{i+k,u})$ is coobservable with respect to L and the extended alphabets $\{a, b, c\}$ and $\{b, d\}$, because now control agent 1 that exerts the control power over the event c is able to distinguish between the strings $s = ab$ after which c should be allowed and $s' = ba$ after which c should be disabled.

VI. Conclusion and Discussion

In this paper, we have shown how to construct a solution of the decentralized control problem (that is, a controllable and coobservable sublanguage of a safety specification) using the results of coordination control. Our approach relies on the notion of conditional decomposability that has recently been studied by the authors. Both modular and coordination control can be applied. Three different sets of conditions are provided that enable to compute a controllable sublanguage of the plant that is coobservable by construction with respect to the observations enriched by communicating coordinator events.

Based on coordination control, we simply compute a controllable sublanguage in a distributed way using the coordinator and the language is by construction conditionally decomposable, hence coobservable with respect to the enriched observable alphabets. This shows a close relationship between coordination control and decentralized control with communication. Recently, we have shown that unlike decomposability and coobservability, conditional decomposability [11] with respect to a large number of local agents (alphabets) can be checked in linear time in the number of agents, although one could expect exponential complexity in the number of agents. This is because the verification of conditional decomposability with respect to n agents can be reduced to n executions of the verification of conditional decomposability with respect to only two agents.

We point out that the influence in the opposite direction is also possible. Coordination control may benefit from decentralized control with communication: individual transitions can be communicated rather than all transitions labeled by an event, which may save space in the communication channel.

Among open directions, it would be nice to develop a more sophisticated approach that would benefit from multilevel hierarchical control. Indeed, although the verification of conditional decomposability is computationally cheap even for a large number of components, the actual size of the coordinator (and its alphabet) might be high and the multilevel hierarchy of coordinators could help. A corresponding approach for the computation of controllable and coobservable sublanguages should then be developed.

We also plan to develop an extension of coordination control that would yield a conditional architecture when applying this extension to inference based decentralized supervisory control. This will require more sophisticated concepts of decomposability and conditional decomposability related to conditional coobservability.

Acknowledgment

This work has been supported by the GAČR grants no. P103/11/0517 and P202/11/P028, by MŠMT grant LH13012 (MUSIC), and by RVO: 67985840.

References

[1] G. Barrett and S. Lafortune. Decentralized supervisory control with communicating controllers. *IEEE Trans. Automat. Control*, 45:1620–1638, 2000.

[2] H. J. Bravo, Antonio E. C. da Cunha, P. N. Pena, R. Malik, and J. E. R. Cury. Generalised verification of the observer property in discrete event systems. In *Proc. of WODES 2012*, pages 337–342, 2012.

[3] C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Springer, 2nd edition, 2008.

[4] H. Chakib and A. Khoumsi. Multi-decision supervisory control: Parallel decentralized architectures cooperating for controlling discrete event systems. *IEEE Trans. Automat. Control*, 56(11):2608–2622, 2011.

[5] L. Feng. *Computationally Efficient Supervisor Design for Discrete-Event Systems*. PhD thesis, University of Toronto, 2007. [Online]. Available at http://www.kth.se/polopoly_fs/1.24026!thesis.zip.

[6] B. Gaudin and H. Marchand. Supervisory control of product and hierarchical discrete event systems. *Eur. J. Control*, 10(2):131–145, 2004.

[7] S. Jiang and R. Kumar. Decentralized control of discrete event systems with specializations to local control and concurrent systems. *IEEE Trans. Syst., Man, Cybern. B*, 30(5):653–660, 2000.

[8] J. Komenda, H. Marchand, and S. Pinchinat. A constructive approach to decentralized supervisory control problems. In *Proc. of DESDes*, pages 111–116, 2006.

[9] J. Komenda, T. Masopust, and J. H. van Schuppen. Synthesis of controllable and normal sublanguages for discrete-event systems using a coordinator. *Systems Control Lett.*, 60(7):492–502, 2011.

[10] J. Komenda, T. Masopust, and J. H. van Schuppen. On algorithms and extensions of coordination control of discrete-event systems. In *Proc. of WODES 2012*, pages 245–250, 2012.

[11] J. Komenda, T. Masopust, and J. H. van Schuppen. On conditional decomposability. *Systems Control Lett.*, 61:1260–1268, 2012.

[12] J. Komenda, T. Masopust, and J. H. van Schuppen. Supervisory control synthesis of discrete-event systems using a coordination scheme. *Automatica*, 48(2):247–254, 2012.

[13] J. Komenda, T. Masopust, and J. H. van Schuppen. Coordination control of discrete-event systems revisited. *http://arxiv.org/abs/1307.4332*, 2013.

[14] R. Kumar and S. Takai. Inference-based ambiguity management in decentralized decision-making: Decentralized control of discrete event systems. *IEEE Trans. Automat. Control*, 52(10):1783–1794, 2007.

[15] S.-H. Lee. *Decentralized Control of Concurrent Discrete-Event Systems*. PhD thesis, The Australian National University, Canberra, Australia, 1998.

[16] S.-H. Lee and K. C. Wong. Structural decentralized control of concurrent discrete-event systems. *Eur. J. Control*, 8:477–491, 2002.

[17] S. L. Ricker and K. Rudie. Know means no: Incorporating knowledge into discrete-event control systems. *IEEE Trans. Automat. Control*, 45(9):1656–1668, 2000.

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

[19] K. Rudie and W. M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. Automat. Control*, 37(11):1692–1708, 1992.

[20] S. Takai and R. Kumar. Synthesis of inference-based decentralized control for discrete event systems. *IEEE Trans. Automat. Control*, 53(2):522–534, 2008.

[21] J. G. Thistle. Undecidability in decentralized supervision. *Systems Control Lett.*, 54(5):503–509, 2005.

[22] S. Tripakis. Undecidable problems of decentralized observation and control on regular languages. *Inform. Process. Lett.*, 90(1):21–28, 2004.

[23] W. M. Wonham. Supervisory control of discrete-event systems. Lecture notes, Department of electrical and computer engineering, University of Toronto, 2012.

[24] T. S. Yoo and S. Lafortune. A general architecture for decentralized supervisory control of discrete-event systems. *Discrete Event Dyn. Syst.*, 12(3):335–377, 2002.

[25] T. S. Yoo and S. Lafortune. Decentralized supervisory control with conditional decisions: Supervisor existence. *IEEE Trans. Automat. Control*, 49(11):1886–1904, 2004.