Multilevel Coordination Control of Modular DES

Jan Komenda, Tomáš Masopust, and Jan H. van Schuppen

Abstract— **A top-down approach to multilevel coordination control is presented along with the corresponding notions of conditional decomposability and conditional controllability. The multilevel structure makes the approach computationally more efficient in comparison with the approach of one central coordinator since fewer events need to be communicated among subsystems. Necessary and sufficient conditions are stated for a specification to be achieved by the proposed top-down approach.**

I. INTRODUCTION

Coordination control of discrete-event systems (DES) has been developed to reduce the combinatorial explosion of the state complexity inherent to supervisory control of large systems. Since purely modular approaches fail in general to guarantee the nonblocking safe behavior, coordination control has been proposed in [7] as a trade-off between purely decentralized (modular) and centralized supervisory control syntheses. The procedure of coordination control consists of the computation of a coordinator for safety and for nonblockingness. Such a coordinator can be seen as an upper layer in the hierarchy, where the low level is the original plant. Coordination control of modular DES combines both horizontal and vertical modularities.

Hierarchical control of DES with complete observations has been studied in the DES literature. Most papers on hierarchical DES address the situation in which one system is abstracted and controlled by another system. In this paper we address the situation where several subsystems at one level are controlled by one subsystem at the next higher level. The important concepts, namely the *observer property* [9] and *output control consistency* (OCC) or its weaker variant *local control consistency* (LCC) [8], are used as sufficient conditions on the abstraction (projection) so that the high-level synthesis of an optimal and nonblocking supervisor for the smaller abstracted plant and specifications is implementable at the low-level (original plant).

Coordination control can be seen as a hierarchical control of a modular plant, where the low level of the hierarchy is the original modular plant and the high level is the coordinator, defined in [5] as the modular plant projected on the coordinator alphabet. However, if there is a large number of local components and a large degree of interactions among local plants, the procedure to compute the coordinator alphabet proposed in [5] yields a too large alphabet. In an extreme case, where all events are shared by some components, the coordinator alphabet becomes the global alphabet. It is because we proposed one central coordinator having in its alphabet all shared events. Clearly, in many practical situations, one central coordinator is not sufficient to decrease the complexity of supervisory control and more sophisticated coordination control architectures should be developed.

In this paper another coordination control architecture is proposed, where one central coordinator is replaced by several coordinators at the second lowest level, which coordinate groups of local subsystems with only limited interactions. The key step in designing this hierarchy is to divide the local plant into several groups such that within each group a very small number of events is shared.

In the proposed top-down approach, control design starts at the top level by computing a coordinator on the high level. Then a coordinator for safety is computed for each group on the lower levels in the top-down manner. The computation then proceeds to the bottom level by computing the coordinator for safety for the low level groups. Finally, at the bottom level, local supervisors must be computed for all groups and all individual subsystems combined with the group coordinators must be computed. No supervisors for safety are needed on the upper levels of the hierarchy, because the specification has been decomposed in the top-down manner with coordinators so that safety is guaranteed.

The paper is organized as follows. Section II recalls the preliminary results from supervisory control with one central coordinator. Section III formulates the top-down approach to multilevel coordination control. Conditional controllability and conditional decomposability conditions for the top-down architecture are formulated in Section IV. In Section V the main result is presented: necessary and sufficient conditions for a specification to be achieved by the top-down approach. Conclusions are given in Section VI.

II. PRELIMINARIES

A string $s \in A^*$ is a *prefix* of $w \in A^*$, denoted by $s \leq w$, if there exists $t \in A^*$ such that $w = st$. The prefix closure $\overline{L} = \{w \in A^* \mid \text{there exists } v \in A^* \text{ such that } wv \in L\}$ of a language $L \subseteq A^*$ is the set of all prefixes of all its elements. A language L is prefix-closed if $L = \overline{L}$.

A *generator* is a structure $G = (Q, A, f, q_0, Q_m)$, where Q is a finite set of *states*, A is a finite *alphabet*, $f : Q \times A \to Q$ is a *partial transition function*, $q_0 \in Q$ is the *initial state*, and $Q_m \subseteq Q$ is the set of *marked states*. As usual, f can be extended to the domain $Q \times A^*$. The *language generated*

J. Komenda and T. Masopust are with Institute of Mathematics, Academy of Sciences of the Czech Republic, Žižkova 22, 616 62 Brno, Czech Rep. komenda@math.cas.cz, masopust@math.cas.cz Part of the research was done when the second author was with the University of Bayreuth, Bayreuth, Germany.

Jan H. van Schuppen is with Van Schuppen Control Research, Gouden Leeuw 143, 1103 KB, Amsterdam, The Netherlands. jan.h.van.schuppen@gmail.com

by G is defined as $L(G) = \{s \in A^* \mid f(q_0,s) \in Q\}$ and the *language marked* by G is defined as $L_m(G) = \{s \in A^* \mid f(q_0,s) \in Q_m\}$. By definition, $L(G)$ is prefix-closed.

A *controlled generator* over an alphabet A is a triple (G, A_c, Γ), where G is a generator over A, $A_c \subseteq A$ is a set of *controllable events*, $A_u = A \setminus A_c$ is the set of *uncontrollable events*, and $\Gamma = \{\gamma \subseteq E \mid A_u \subseteq \gamma\}$ is the *set of control patterns*. A *supervisor* for a controlled generator (G, A_c, Γ) is a map $S : L(G) \to \Gamma$. The *closed-loop system* associated with controlled generator (G, A_c, Γ) and supervisor S is defined as the minimal language $L(S/G)$ such that $\varepsilon \in L(S/G)$ and, for any $s \in L(S/G)$ with $sa \in L(G)$ and $a \in S(s)$, sa belongs to $L(S/G)$. The marked language of the closed-loop system is defined as $L_m(S/G) = L(S/G) \cap L_m(G)$. If the closed-loop system is nonblocking, that is $\overline{L_m(S/G)} = L(S/G)$, supervisor S is called *nonblocking*.

Let $L_m, L \subseteq A^*$ be languages, where L is prefix-closed. A language $K \subseteq A^*$ is *controllable* with respect to L and A_u if $\overline{K}A_u \cap L \subseteq \overline{K}$. Moreover, K is L_m-closed if $K = \overline{K} \cap L_m$.

A *projection* $P : A^* \to B^*$, for $B \subseteq A$, is a homomorphism defined as $P(a) = \varepsilon$, for $a \in A \setminus B$, and $P(a) = a$, for $a \in B$. The *inverse image* of P, denoted by $P^{-1} : B^* \to 2^{A^*}$, is defined as $P^{-1}(w) = \{s \in A^* \mid P(s) = w\}$. These definitions can be extended to languages. For alphabets A_i, A_j, $A_\ell \subseteq A$, we use P_ℓ^{i+j} to denote the projection from $(A_i \cup A_j)^*$ to A_ℓ^*. If $A_i \cup A_j = A$, we simply write P_ℓ. Moreover, $A_{i,u} = A_i \cap A_u$ denotes the sets of locally uncontrollable events. For a generator G and a projection P, $P(G)$ denotes the minimal generator such that $L_m(P(G)) = P(L_m(G))$ and $L(P(G)) = P(L(G))$. The reader is referred to [1], [10] for a construction.

Let G be a generator over an alphabet A. Given a specification $K \subseteq L_m(G)$, the aim of supervisory control is to find a nonblocking supervisor S such that $L_m(S/G) = K$. Such a supervisor exists if and only if K is controllable with respect to $L(G)$ and A_u and $L_m(G)$-closed, see [1], [10].

The synchronous product of languages $L_i \subseteq A_i^*$, $i = 1, \ldots, n$, is defined as $\|_{i=1}^n L_i = \cap_{i=1}^n P_i^{-1}(L_i) \subseteq A^*$, where $A = \cup_{i=1}^n A_i$ and $P_i : A^* \to A_i^*$ are projections to local alphabets. In terms of generators G_i, it is known that $L(\|_{i=1}^n G_i) = \|_{i=1}^n L(G_i)$ and $L_m(\|_{i=1}^n G_i) = \|_{i=1}^n L_m(G_i)$ (see [1] for more details). Languages K and L are *synchronously nonconflicting* if $\overline{K \| L} = \overline{K} \| \overline{L}$.

A projection $Q : A^* \to B^*$ is an *L-observer* for a language $L \subseteq A^*$ if, for every $t \in Q(L)$ and $s \in \overline{L}$, $Q(s) \le t$ implies that there is $u \in A^*$ such that $su \in L$ and $Q(su) = t$ [9].

Now we recall the basic notion of coordination control.

Definition 1 (Conditional decomposability): A language K over $\cup_{i=1}^n A_i$ is *conditionally decomposable with respect to* $(A_i)_{i=1}^n$ *and* A_k, where $\cup_{1\le i,j\le n}^{i\ne j}(A_i \cap A_j) \subseteq A_k \subseteq \cup_{j=1}^n A_j$, if

$$K = P_{1+k}(K) \parallel P_{2+k}(K) \parallel \ldots \parallel P_{n+k}(K)$$

for projections P_{i+k} from $\cup_{j=1}^n A_j$ to $A_i \cup A_k$, $i = 1, \ldots, n$. $\triangleleft$

Alphabet A_k is referred to as a coordinator alphabet and satisfies the *conditional independence* property, namely A_k includes all shared events: $\cup_{1\le i,j\le n}^{i\ne j}(A_i \cap A_j) \subseteq A_k$. It holds that if K is a parallel composition of n languages (over the required alphabets), then it is conditionally decomposable.

Lemma 2 (Lemma 2 in [4]): A language K over $\cup_{i=1}^n A_i$ is conditionally decomposable with respect to alphabets $(A_i)_{i=1}^n$ and A_k if and only if there exist languages $M_{i+k} \subseteq A_{i+k}^*$, $i = 1, \ldots, n$, such that $K = \|_{i=1}^n M_{i+k}$. ∎

Now we recall the main result of coordination control with one central coordinator. The problem of coordination control is as follows.

Problem 3: Given generators G_1 and G_2 over alphabets A_1 and A_2, respectively, and a coordinator G_k over A_k, where $A_1 \cap A_2 \subseteq A_k \subseteq A_1 \cup A_2$. Let $K \subseteq L_m(G_1 \| G_2 \| G_k)$ be a specification that is conditionally decomposable with respect to A_1, A_2, A_k. The problem of coordination control is to synthesize nonblocking supervisors S_1, S_2, S_k for the respective generators so that the closed-loop system with the coordinator satisfies

$$L_m(S_1/[G_1 \| (S_k/G_k)]) \parallel L_m(S_2/[G_2 \| (S_k/G_k)]) = K. \quad \triangleleft$$

The idea of coordination control is to first construct a supervisor S_k such that the closed-loop system $L(S_k/G_k)$ satisfies the "coordinator part" of the specification given by $P_k(K)$ and then local supervisors S_i, $i = 1, 2$, for $G_i \| (S_k/G_k)$ such that the closed-loop system $L(S_i/[G_i \| (S_k/G_k)])$ satisfy the corresponding parts of the specification given by $P_{i+k}(K)$.

Conditional controllability along with conditional decomposability form an equivalent condition for a language to be achieved by the closed-loop system within our coordination control architecture, cf. Theorem 5 below.

Definition 4: A language $K \subseteq L(G_1 \| G_2 \| G_k)$ is *conditionally controllable* for generators G_1, G_2, G_k and uncontrollable alphabets $A_{1,u}$, $A_{2,u}$, $A_{k,u}$ if

1) $P_k(K)$ is controllable with respect to $L(G_k)$ and $A_{k,u}$,
2) $P_{i+k}(K)$ is controllable with respect to $L(G_i) \parallel \overline{P_k(K)}$ and $A_{i+k,u}$,

where $A_{i+k,u} = (A_i \cup A_k) \cap A_u$, for $i = 1, 2$. $\triangleleft$

Recall that every conditionally controllable and conditionally decomposable language is controllable, cf. [3, Proposition 4]. The main existential result is the following.

Theorem 5 (Theorem 6 in [5]): Consider the setting of Problem 3. There exist nonblocking supervisors S_1, S_2, S_k such that $L(S_1/[G_1 \| (S_k/G_k)]) \parallel L(S_2/[G_2 \| (S_k/G_k)]) = K$ if and only if K is conditionally controllable with respect to generators G_1, G_2, G_k and alphabets $A_{1,u}$, $A_{2,u}$, $A_{k,u}$. ∎

III. Multilevel coordination control

In this section we study a computationally efficient approach to supervisory control of a large modular DES given by a synchronous product of generators. The single-coordinator approach of [5] is replaced by several coordinators on different levels. The first step is to divide local subsystems into groups of subsystems on the lowest level. Each group then has its own coordinator. Here we assume that the organization of subsystems into groups is given by the system designer. A criterion for this organization can be the number of shared events within groups of subsystems, which makes this organization sometimes obvious from the geographical distribution of subsystems. The motivation for this division into several groups is that it is typically needed to include many events in the coordinator alphabet to make the specification language conditionally decomposable, especially in the case of a large number of subsystems. Instead of adding all events that have to be communicated into a central coordinator alphabet it is more efficient if each coordinator event is communicated only within some group(s) of subsystems, which amounts to having different coordinators for different groups while dividing the coordinator alphabet into different subsets communicated among subsystems within given group(s).

Let $G = G_1 \| G_2 \| \ldots \| G_n$ and assume that local generators are divided into m groups. We change the indexing so that the first group is formed by generators $G_1, \ldots, G_{i_1}$, the second group by $G_{i_1+1}, \ldots, G_{i_2}$, and so forth, i.e. the m-th group is formed by $G_{i_{m-1}+1}, \ldots, G_{i_m}$, where $1 \le i_1 \le i_2 \le \cdots \le i_m = n$. Recall that the synchronous product is associative and commutative, hence we can organize the subsystems in an arbitrary way. Denote the indexes of generators of the j-th group by I_j, i.e. $I_j = \{i_{j-1}+1, i_{j-1}+2, \ldots, i_j\}$, for $j = 1, \ldots, m$ where $i_0 = 0$. Similarly, we assume that the groups of subsystems $I_1, \ldots, I_m$ are organized into ℓ larger groups $J_1, \ldots, J_\ell$ with $\ell \le m$, and so on. For simplicity, however, we consider in this paper the case $\ell = 1$, that is, we have only two levels of organization, where on the second level one obtains the complete system $G_1 \| \ldots \| G_n$. In other words, we consider $J_1 = \{I_1, \ldots, I_m\}$ meaning that $\|_{i \in I}^{I \in J_1} G_i = G_1 \| \ldots \| G_n$. However, in the general multilevel case not considered in this paper, the groups I_j can be further gathered up into larger groups $J_1, \ldots, J_\ell$ with $\ell \le m$ on the higher level and so forth.

An important aspect is to propose a criterion for such a hierarchical structure of subsystems. We do not propose it in a formal way, but only provide a hint on how to build such a hierarchical structure. The idea is to bundle subsets of subsystems with strong interactions at the lowest level of the multilevel structure. In the ideal situation the automata formed by products of generators from different low level groups have no shared events. This intuition can be made mathematical by associating the subset with a square matrix with the number of shared events between the subsystems in a row and a column and try to find a permutation and a block matrix structure such that the maximum of shared events is situated in the diagonal blocks, while off-diagonal blocks contain very small numbers (ideally zero matrices).

Finally, denote by $A_{sh,j}$ the set of shared events of generators $G_{i_{j-1}+1}, \ldots, G_{i_j}$ of group I_j, i.e.

$$A_{sh,j} = \bigcup_{k,\ell \in I_j}^{k \neq \ell} (A_k \cap A_\ell)\,.$$

Unlike in central coordination, at the low level there are m low-level coordinators $G_{k_1}, \ldots, G_{k_m}$, one for each group of subsystems. The situation is depicted in Fig. 1. The notation

$$A_{I_r} = \bigcup_{i \in I_r} A_i$$

is used in the paper. Here P_{I_r} denotes the projection $P_{I_r} : A^* \to A_{I_r}^*$. On the highest level there is one central coordinator denoted by G_k over the alphabet A_k that coordinates the m groups of subsystems. We hope that the notation for projection $P_{I_r+k} : A^* \to (A_{I_r} \cup A_k)^*$ is now self-explanatory. Again, the high-level coordinator should contain all shared events, in this case all events shared by the groups of subsystems denoted by

$$A_{sh} = \bigcup_{k,\ell \in \{1,\ldots,m\}}^{k \neq l} (A_{I_k} \cap A_{I_\ell})\,.$$

Hence, $A_{sh} \subseteq A_k$, which is later referred to as the *conditional independence* assumption.

Note that, in general, A_{sh} contains fewer events than all shared events among all subsystems. In the special case, where events are only shared by subsystems within each groups, we have $A_{sh} = \emptyset$. This confirms the intuition that it is the best to leave the maximum interaction among subsystems to be handled at the lowest level. Note that although no high-level coordination for nonblocking is needed at all (because subsystems on disjoint alphabets can be supervised in a modular way without the blocking problem), a high-level coordination for safety is still needed whenever the specification language is not decomposable with respect to high-level alphabets $A_{I_1}, \ldots, A_{I_m}$.

IV. Control synthesis - Top-down approach

Once the organization of subsystems into groups is fixed, we study the multilevel coordination control synthesis. A notion of two-level conditional decomposability is now introduced. In what follows only prefix-closed specification languages are considered. The alphabet $A_k \subseteq A$ (corresponding to the high-level coordinator) is assumed to satisfy the conditional independence property $A_{sh} \subseteq A_k$ as well as alphabets $A_{k_r} \subseteq A_{I_r}$, $r = 1, \ldots, m$, are assumed to satisfy the conditional independence property $A_{sh,r} \subseteq A_{k,r}$ at the local group.

Definition 6 (Two-level conditional decomposability): A language $K \subseteq A^*$ is called *two-level conditionally decomposable* with respect to alphabets $A_1, \ldots, A_n$, high-level coordinator alphabet A_k, and low-level coordinator alphabets $A_{k_1}, \ldots A_{k_m}$ if

$$K = \|_{r=1}^m P_{I_r+k}(K) \quad \text{and} \quad P_{I_r+k}(K) = \|_{j \in I_r} P_{j+k_r+k}(K)$$

for $r = 1, \ldots, m$. ◁

Recall that P_{j+k_r+k} stands for the projection from A^* to $A_{j+k_r+k}^* = (A_j \cup A_{k_r} \cup A_k)^*$. For the set of second equations, the specification of the group over $A_{I_r} \cup A_k$ is not in general decomposable into individual alphabets of group I_r enriched with corresponding low-level coordinator events A_{k_r} because the high-level coordinator events A_k might be from alphabets corresponding to different groups. Therefore, we have to include the global coordinator events as well to have a meaningful equation comparing languages over the same alphabets on both sides.

The list of coordinator alphabets A_k, $A_{k_1}, \ldots A_{k_m}$ is omitted from the expression if it is clear from the context. Note that the existence of coordinator alphabets $A_{k_1}, \ldots A_{k_m}$ such that

$$K = \left(\|_{j_1 \in I_1} P_{j_1+k_1+k}(K)\right) \| \ldots \| \left(\|_{j_m \in I_m} P_{j_m+k_m+k}\right)$$

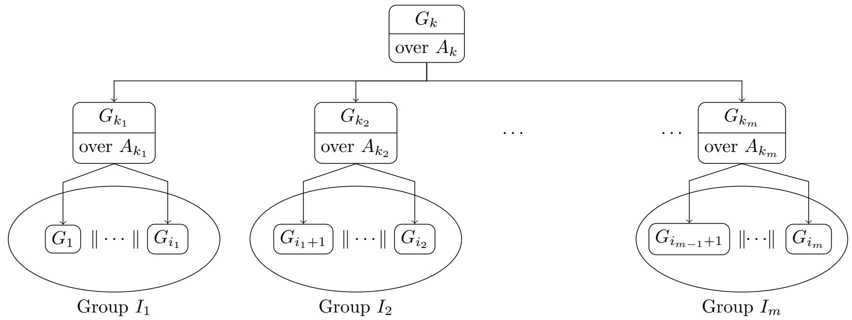

Fig. 1. Multilevel architecture

implies that $K = \|_{i=1}^n P_{i+h}(K)$ with $A_h = A_{k_1} \cup \cdots \cup A_{k_m} \cup A_k$. This is because for this choice of A_h we have in fact that $P_{i+h}^{-1} P_{i+h}(K) \subseteq P_{i+k_j+k}^{-1} P_{i+k_j+k}(K)$, for $j \in \{1,\ldots,m\}$. This means that two-level conditional decomposability implies (standard) conditional decomposability, but with respect to larger alphabets. Here the idea of two-level decomposability is easily seen: instead of communicating all coordinator events via a central coordinator, it is more advantageous to communicate different parts of A_k, namely $A_{k_1},\ldots,A_{k_m}$, within the respective groups of subsystems I_i via the corresponding "group" coordinators G_{k_i}, for $i = 1,\ldots,m$.

On the other hand the following property holds true.

Proposition 7: If a language $K \subseteq A^*$ is conditionally decomposable with respect to alphabets $(A_i)_{i=1}^n$ and A_h, then it is two-level conditionally decomposable with respect to alphabets $(A_i)_{i=1}^n$ and coordinator alphabets $A_{k_1} = \cdots = A_{k_m} = A_k = A_h$, for any $m > 1$. ■

However, the opposite does not hold true.

Example 8: Let $K \subseteq \{a_1,a_2,a_3,a_4\}^*$ be a language given as a parallel composition of languages $K_{12} \subseteq \{a_1,a_2\}^*$ and $K_{34} \subseteq \{a_3,a_4\}^*$ depicted in Fig. 2. By Lemma 2, K is conditionally decomposable with respect to alphabets $\{a_1,a_2\}$ and $\{a_3,a_4\}$. Moreover, $K_{12} = P_{1+2}(K)$ and $K_{34} = P_{3+4}(K)$. Hence, $K = P_{1+2}(K) \| P_{3+4}(K)$, which means that in Definition 6 we can choose $A_k = \emptyset$. Then we take $A_{k_1} = \{a_1\}$ and $A_{k_2} = \{a_4\}$ to guarantee that $K_{12} = P_{1+k_1}(K_{12}) \| P_{2+k_1}(K_{12})$ and $K_{34} = P_{3+k_2}(K_{34}) \| P_{4+k_2}(K_{34})$. Finally, to make K conditionally decomposable with respect to $(\{a_i\})_{i=1}^4$ and $A_{k'}$, $A_{k'}$ must contain at least one of a_1 and a_2, and one of a_3 and a_4, hence $|A_{k'}| \geq 2$, whereas $|A_{k_1}| = |A_{k_2}| = 1$. ◁

Communications among local generators are reduced, because unlike the original concept of conditional decomposability, where all events A_k are communicated among all local agents via the coordinator, the events that need to be communicated are now divided into groups of events associated to a group of subsystems and their coordinators and the events are communicated among local subsystems belonging to a given group via the corresponding coordinator. Moreover, in view of the previous result, it is often the case that low-level coordinators $A_{k_1},\ldots,A_{k_m}$ are able to operate on smaller alphabets than the full A_k. In general, A_k can be distributed into $A_{k_i} \subseteq A_k$, $i = 1,\ldots,m$, with $\cup_{i=1}^m A_{k_i} = A_k$.

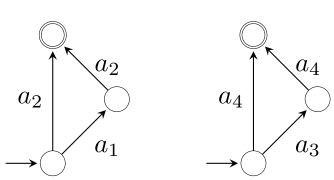

Fig. 2. Generators of languages K_{12} and K_{34}, respectively

Example 9: In this example we consider four generators G_1, …, G_4 over the alphabets A_1, …, A_4, respectively, and their synchronous product $G = G_1 \| \ldots \| G_4$. On the low (system) level we divide the four generators into two groups $I_1 = \{1,2\}$ and $I_2 = \{3,4\}$. There are low-level coordinators G_{k_1} and G_{k_2} coordinating subsystems $G_1 \| G_2$ and $G_3 \| G_4$, respectively. It is assumed that the specification K is two-level conditionally decomposable with respect to the high-level coordinator alphabet A_k, and low-level coordinator alphabets A_{k_1}, …, A_{k_m}, that is, $K = P_{1+2+k}(K) \| P_{3+4+k}(K)$, $P_{1+2+k}(K) = P_{1+k_1+k}(K) \| P_{2+k_1+k}(K)$, and $P_{3+4+k}(K) = P_{3+k_2+k}(K) \| P_{4+k_2+k}(K)$.

Multilevel coordination control architecture is defined later, but we sketch it now in this example to facilitate the formal presentation of Problem 10 below. For each low level group of coordinators combined with the high level coordinator (note that parts of the specification alphabets $A_k \cup A_{k_i}$, $i = 1,2$, must be considered jointly), there must be supervisors S_{k_1} for $G_k \| G_{k_1}$ and S_{k_2} for $G_k \| G_{k_2}$ that impose the corresponding part of the specification.

For local subsystems combined with the supervised coordinators there are local supervisors S_i, for $i = 1,2,3,4$. Namely, S_1 supervises the new plant $G_1 \| (S_{k_1}/G_k \| G_{k_1})$ with the resulting closed-loop system $L(S_1/(G_1 \| (S_{k_1}/G_k \| G_{k_1})))$. Similarly, S_2 supervises $G_2 \| (S_{k_1}/G_k \| G_{k_1})$, S_3 supervises $G_3 \| (S_{k_2}/G_k \| G_{k_2})$, and S_4 supervises $G_4 \| (S_{k_2}/G_k \| G_{k_2})$.

On the high level, there is only a high-level coordinator G_k that plays an auxiliary role in decomposing K on the high level. There is no need for any supervisor on the high level: neither for G_k nor for the combined high-level plant. Otherwise stated, all follow from two-level conditional decomposability combined with two-level conditional controllability presented below. Hence, the overall two-level coordinated system is the composition

$$S_1/[G_1 \parallel (S_{k_1}/G_k \| G_{k_1})] \ \parallel \ S_2/[G_2 \parallel (S_{k_1}/G_k \| G_{k_1})] \ \parallel$$
$$S_3/[G_3 \parallel (S_{k_2}/G_k \| G_{k_2})] \ \parallel \ S_4/[G_4 \parallel (S_{k_2}/G_k \| G_{k_2})]. \qquad \triangleleft$$

The two-level coordination control problem of modular DES is formulated below.

Problem 10 (Two-level coordination control problem): Consider generators $G_1,\dots,G_n$ over alphabets $A_1,\dots,A_n$, respectively, and their synchronous product $G = G_1 \| \dots \| G_n$ along with the two-level hierarchical structure of subsystems organized into groups $I_j = \{i_{j-1}+1, i_{j-1}+2,\dots,i_j\}$, $j = 1,\dots,m \le n$, on the low level. The synchronous products of generators from these groups then represent the m high-level systems $\|_{i\in I_j} G_i$, $j = 1,\dots,m$. It is assumed that the specification K is prefix-closed and two-level conditionally decomposable with respect to local alphabets $A_1,\dots,A_n$, high-level coordinator alphabet A_k, and low-level coordinator alphabets $A_{k_1},\dots A_{k_m}$. The two-level structure of coordinators is associated to the above organization of subsystems into groups in a natural way. Namely, on the low level coordinator G_{k_j} is associated to the group of subsystems $\{G_i \mid i \in I_j\}$, $j = 1,\dots,m$. On the high level, a unique (central) coordinator is denoted by G_k. The aim of the two-level coordination control synthesis is to determine supervisors S_i, $i \in I_j$, within any group of low-level systems $\{G_i \mid i \in I_j\}$, $j = 1,\dots,m$, and supervisors for low-level coordinators combined with the high-level coordinator S_{k_j}, $j = 1,\dots,m$, such that the specification is met by the closed-loop system. The overall two-level coordinated and supervised closed-loop system is given by

$$\|_{j=1}^{m} \|_{i\in I_j} L(S_i/[G_i \parallel (S_{k_j}/G_k \| G_{k_j})]) . \qquad \triangleleft$$

In the statement of the problem, we have mentioned the notion of a coordinator. Given a specification K, the coordinator G_{k_j} of the j-th group of subsystems $\{G_i \mid i \in I_j\}$ is computed as follows.

1) Set $A_{k_j} = A_{sh,j} = \bigcup_{k,\ell\in I_j}^{k\neq\ell} (A_k \cap A_\ell)$ to be the set of all shared events of systems from the group I_j.
2) Extend A_{k_j} so that $P_{I_r+k}(K)$ is conditional decomposable with respect to $(A_i)_{i\in I_j}$ and A_{k_j}, for instance using a method described in [4].
3) Let coordinator $G_{k_j} = \|_{i\in I_j} P_{k_j}(G_i)$.

The high-level coordinator G_k is computed in a similar way as G_{k_j}, but instead of the low-level groups, all local subsystems are used, i.e. $G_k = \|_{i=1}^{n} P_k(G_i)$.

Since the only known condition ensuring that the projected generator is smaller than the original one is the observer property [9] we might need to further extend alphabet A_{k_j} so that projection P_{k_j} is an $L(G_i)$-observer, for any $i \in I_j$.

Note that the blocking issue is not considered in this paper, because the specification is assumed to be prefix-closed. However, we have recently solved the blocking issue by proposing coordinators for nonblockingness. These coordinators are computed in a different way than the coordinators for safety considered in this paper and defined above, cf. [6]. The extension of coordinators for nonblockingness from one-level coordination control to two-level coordination control is fairly simple once the framework is established.

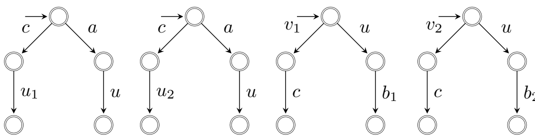

Fig. 3. Generators $G_1,\dots,G_4$

The central notion in the coordination control approach is played by the concept of conditional controllability introduced in [7] and later studied in [2], [5], [3]. In this paper, we extend this notion as follows.

Definition 11: Consider the setting and notation of Problem 10 and let G_k be a coordinator. A language $K \subseteq L(\|_{i=1}^{n} G_i \parallel G_k)$ is *two-level conditionally controllable* with respect to generators $G_1,\dots,G_n$, local alphabets $A_1,\dots,A_n$, high-level coordinator alphabet A_k, low-level coordinator alphabets $A_{k_1},\dots A_{k_m}$, and uncontrollable alphabet A_u if

1) $P_{k_j+k}(K)$ is controllable with respect to $L(G_{k_j} \| G_k)$ and $A_{k_j+k,u}$,
2) for $j = 1,\dots,m$ and $i \in I_j$, $P_{i+k+k_j}(K)$ is controllable with respect to $L(G_i) \parallel P_{k_j+k}(K)$ and $A_{i+k_j+k,u}$. $\triangleleft$

V. Existence of Supervisors

In this section, the main existential result of top-down multilevel coordination control approach is presented. We start at the top level by decomposing the specification according to the distribution of alphabets. Then a similar decomposition is computed at the lower level. The actual computation of coordinators and supervisors is made at the lowest level. No further computation is needed on the higher levels, because the overall specification is satisfied by construction.

Theorem 12: Consider the setting of Problem 10 (in particular K is two-level conditionally decomposable with respect to local alphabets $A_1,\dots,A_n$, high-level coordinator alphabet A_k, and low-level coordinator alphabets $A_{k_1},\dots A_{k_m}$). There exist supervisors for low-level systems S_i, $i \in I_j$, within any group of low-level systems $\{G_i \mid i \in I_j\}$, $j = 1,\dots,m$, and supervisors S_{k_j}, $j = 1,\dots,m$, for low-level coordinators combined with the high-level coordinator, such that

$$\|_{j=1}^{m} \|_{i\in I_j} L(S_i/G_i \parallel (S_{k_j}/G_k \| G_{k_j})) = K \qquad (1)$$

if and only if K is two-level conditionally controllable with respect to generators and alphabets listed in Definition 11. ∎

If K fails to be two-level conditional controllable, a sublanguage of K that is conditional controllable is computed. Fortunately, similarly to one-level conditional controllability, two-level conditional controllability is preserved by language unions, whence the supremal two-level conditional controllable sublanguage always exists.

Example 13: Example 9 can be continued with a concrete modular system. Let $A_1 = \{a,c,u,u_1\}$, $A_2 = \{a,c,u,u_2\}$, $A_3 = \{b_1,c,u,v_1\}$, and $A_4 = \{b_2,c,u,v_2\}$ where $G_1,\dots,G_4$ are defined in Fig. 3, and $A_u = \{u,u_1,u_2\}$. The specification K is defined in Fig. 4. Following the procedure for the top-down computation scheme we need to check if K is

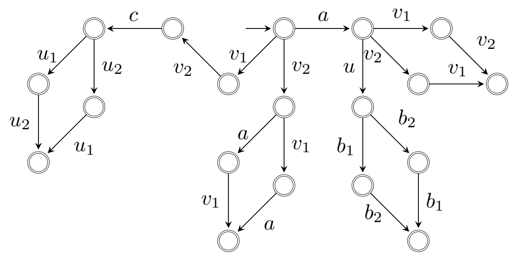

Fig. 4. Generator for the specification K

two-level conditionally decomposable. It appears that we have to extend the alphabets of shared events to make this condition hold. First of all, by choosing $A_k = \{a,c,u\}$, i.e. by extending the high level shared alphabet $A_{sh} = (A_1 \cup A_2) \cap (A_3 \cup A_4)$ by event a we get $K = P_{1+2+k}(K) \| P_{3+4+k}(K)$. The corresponding high-level coordinator is then given by $L_k = P_k(L) = \{\varepsilon, c, a, au\}$. The low-level conditions of two-level conditionally decomposability require to find low-level coordination alphabets A_{k_1} and A_{k_2}. There is no need to extend $A_{sh,1} = A_1 \cap A_2$, because $P_{1+2+k}(K) = P_{1+k}(K) \| P_{2+k}(K)$ is actually decomposable with respect to alphabets $A_1 = A_1 \cup A_k$ and $A_2 = A_2 \cup A_k$. Hence, $A_{k_1} = A_1 \cap A_2 = \{a,c\}$. On the other hand, $P_{3+4+k}(K)$ is not decomposable with respect to A_{3+k} and A_{4+k}. For $A_{k_2} = A_{sh,2} \cup \{v_1\} = \{c,u,v_1\}$ we have $P_{3+4+k}(K) = P_{3+k+k_2}(K) \| P_{4+k+k_2}(K)$, i.e. $P_{3+4+k}(K)$ is conditionally decomposable with respect to alphabets A_3, A_4, and $A_{k+k_2} = A_k \cup A_{k_2}$.

Once we have conditionally decomposed the global specification in a top-down manner for coordinator alphabets A_k, A_{k_1} and A_{k_2}, we can start the computation at the bottom level. It can be checked that the specification K is two-level conditionally controllable with respect to the same coordinator alphabets (no further extension is needed). We start with the language $P_{1+2+k}(K) = P_{1+k}(K) \| P_{2+k}(K)$. Since $P_{i+k}(K) = P_i(K) = L_i$, $i = 1,2$, there is no need to compute supervisors and coordinators for the group I_1. For the group I_2, the low-level coordinator is given by $L_{k_2} = P_{k_2}(L_3 \| L_4) = \{\varepsilon, u, v_1, v_1 c\}$. $P_{3+k+k_2}(K)$ has to be imposed for the part of the global plant $L_3 \| L_{k_2} \| L_k$. Fortunately, $P_{3+k+k_2}(K)$ is controllable with respect to the language $L_3 \| L_{k_2} \| L_k$ and, hence, $\sup \mathrm{C}(P_{3+k+k_2}(K), L_3 \| L_{k_2} \| L_k, A_{3+k+k_2,u}) = P_{3+k+k_2}(K)$.

Indeed, it suffices to disable controllable event a after v_1 has occurred. Languages $P_{3+k+k_2}(K)$ and $L_3 \| L_{k_2} \| L_k$ are depicted in Fig. 5. Similarly, $P_{4+k+k_2}(K)$ is controllable with respect to $L_4 \| L_{k_2} \| L_k$ and no computation of the supremal controllable sublanguage is needed, see Fig. 6. Here, it also suffices to disable a after v_1 has occurred.

It can be checked that the overall closed-loop language is $P_{1+k}(K) \| P_{2+k}(K) \| P_{3+k+k_2}(K) \| P_{4+k+k_2}(K) = K$, in accordance with the two-level conditional decomposability and two-level conditional controllability of K. $\triangleleft$

VI. Concluding remarks

In a future publication, it is our plan to apply multilevel coordination control to modular control of DES with communicating supervisors. This way we obtain interesting communication protocols among local supervisors via coordinators for different groups of subsystems.

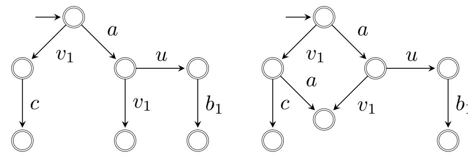

Fig. 5. $P_{3+k+k_2}(K)$ and $L_3 \| L_{k_2} \| L_k$

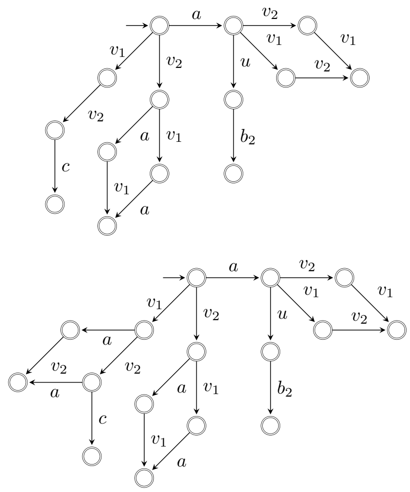

Fig. 6. $P_{4+k+k_2}(K)$ and $L_4 \| L_{k_2} \| L_k$

VII. Acknowledgments

The authors gratefully acknowledge comments and suggestions of the anonymous referees. The research was supported by GAČR grants P103/11/0517 and P202/11/P028, by MŠMT grant LH13012 (MUSIC), and by RVO: 67985840.

References

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

[2] 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.*, vol. 60, no. 7, pp. 492–502, 2011.

[3] ——, "On algorithms and extensions of coordination control of discrete-event systems," in *WODES*, Guadalajara, Mexico, 2012, pp. 245–250.

[4] ——, "On conditional decomposability," *Systems Control Lett.*, vol. 61, no. 12, pp. 1260–1268, 2012.

[5] ——, "Supervisory control synthesis of discrete-event systems using a coordination scheme," *Automatica*, vol. 48, no. 2, pp. 247–254, 2012.

[6] ——, "Coordination control of discrete-event systems revisited," *http://arxiv.org/abs/1307.4332*, 2013.

[7] J. Komenda and J. H. van Schuppen, "Coordination control of discrete event systems," in *WODES*, Gothenburg, Sweden, 2008, pp. 9–15.

[8] K. Schmidt and C. Breindl, "On maximal permissiveness of hierarchical and modular supervisory control approaches for discrete event systems," in *WODES*, Gothenburg, Sweden, 2008, pp. 462–467.

[9] K. Wong and W. Wonham, "Hierarchical control of discrete-event systems," *Discrete Event Dyn. Syst.*, vol. 6, no. 3, pp. 241–273, 1996.

[10] W. M. Wonham, "Supervisory control of discrete-event systems," 2012, lecture notes, University of Toronto, [Online]. Available at http://www.control.utoronto.ca/DES/.