

Modular Supervisory Control with General Indecomposable Specification Languages

J. Komenda

*Institute of Mathematics, Czech Academy of Sciences,
Brno Branch, Zizkova 22, 616 62 Brno, Czech Republic*

komenda@ipm.cz

B. Gaudin

*IRISA, Université de Rennes 1,
Campus de Rennes 1, 35042 Rennes, France*

Benoit.Gaudin@irisa.fr

J.H. van Schuppen

*CWI, P.O. Box 94079,
1090 GB Amsterdam, The Netherlands*

J.H.van.Schuppen@cwil.nl

H. Marchand

*IRISA, INRIA Rennes, Campus de Rennes 1,
35042 Rennes, France*

Herve.Marchand@irisa.fr

Abstract—Modular supervisory control of discrete-event systems (DES), where the global DES is composed of local components that run concurrently, is considered. For supervisory control of large-scale modular DES the possibility of performing control-related computations locally (in components) is of utmost importance to computational complexity.

Recently we have treated the case, where the specification language is decomposable into local specification languages and is included in the (global) plant language. In this paper the case of general specification languages that are neither necessarily decomposable nor contained in the global plant language is studied. Sufficient conditions are found under which any manipulation with the global plant is avoided for the computation of supremal controllable sublanguages of (global) indecomposable specification languages.

Keywords— Modular control of discrete-event systems, Partial controllability, Coalgebra, Supremal controllable sublanguages

I. INTRODUCTION

Supervisory control of modular DES (also called concurrent DES) is considered. Control of DES represented by finite automata have been introduced by P.J. Ramadge and W.M. Wonham (see e.g. [10]). Large scale modular DES are typically composed of a large number of relatively small (in size) local DES that run concurrently (in parallel). The global system is formed as a synchronous product of these local components with synchronization on shared actions.

The input alphabets of the local components were identical in the first papers on this topic ([14]). In [11] and [2] quite a restrictive condition is imposed on events shared by several local alphabets: they must be controllable for all subsystems. This assumption has been generalized recently in [12] to the condition that the shared events must have the same control status for all subsystems that share a particular event. This rather general assumption together with general specification languages for the global plant (i.e. those that are not decomposable into local specification

languages) are considered in this paper. Indecomposable specification languages have been first considered in [2] under the conditions that all shared events are controllable.

Our attention is restricted to modular control synthesis without blocking as the blocking issue requires different concepts and methods. Preliminary results on coalgebra and coinduction can be found in the appendix. It is highly recommended that the reader goes through the appendix before starting to read Section 3. Section 2 is devoted to the presentation and motivation of the main problem studied in this paper. Section 3 is devoted to our main results. Two novel sufficient conditions (a specification dependent and a structural one) are presented under which optimal modular control synthesis with complete observations is possible without building the global plant and without a loss of global optimality.

II. PROBLEM STATEMENT

It is very well known that not every specification language can be achieved by a supervisory controller [10]. In the case of basic supervisory control problems with complete observations we look for optimal safe approximations that can be formulated in terms of supremal controllable sublanguages of specification languages. Therefore algorithms for computation of supremal controllable sublanguages have been derived [10].

In modular control the concurrent behavior of local subplants (partial automata) G_1, \dots, G_n is considered. The notation $\mathbb{Z}_n = \{1, 2, \dots, n\}$ is used. The global plant is the synchronous product $G = \parallel_{i=1}^n G_i$. Although the computational complexity of the algorithm for supremal controllable sublanguages is satisfactory (unlike several other problems of supervisory control), in the case of large modular DES there is an exponential blow up of the computational complexity in terms of the number of local components. In order to cope with this issue, modular structure should be exploited.

In this paper we face moreover the problem of indecomposable specification languages, i.e. we do not have local specification languages. Nevertheless, as is shown in [2], it is still possible to exploit the modular structure of the plant and to avoid the manipulation with the global plant. On the other hand, the solution proposed in [2] relies on rather restrictive structural conditions, where all shared events are required to be controllable for all subsystems that share a particular event. We want to weaken this condition, while still preserving the possibility of 'local' computation, i.e. without having to manipulate with the global plant language. The problem can be formulated as follows:

Problem 2.1: Propose an algorithm for computation of supremal controllable sublanguages in the context of modular DES with indecomposable specification languages that avoids the manipulation with the global plant language.

Since a methodology to cope with general specifications has already been introduced in [2], the main scope of the paper is to generalize the results of [2] in two directions: 1) leave out the structural condition that all shared events are controllable and 2) leave out the restriction on the indecomposable specification by finding a structural (specification independent) conditions under which the methodology proposed in [2] can still be applied.

III. SUPREMAL CONTROLLABLE SUBLANGUAGES OF GENERAL SPECIFICATION LANGUAGES.

As customary in modular supervisory control, local alphabets of the local sub-plants, denoted by A_i , $i \in \mathbb{Z}_n$ and not necessarily pairwise disjoint, are such that $A_i = A_{iu} \cup A_{ic}$, where A_{iu} stands for the subset of locally uncontrollable events and A_{ic} stands for the subset of locally controllable events. We assume that

$$\forall i \neq j, A_{iu} \cap A_j = A_i \cap A_{ju} \quad (1)$$

This assumption originally stemming from [12] means that the events shared by two local subsystems must have the same control status for both controllers associated to these subsystems. We denote by $A_c = \cup_{i=1}^n A_{ic}$ and $A_u = A \setminus A_c$ the subsets of globally controllable and globally uncontrollable events, respectively. We still have $A_u = \cup_{i=1}^n A_{iu}$ due to the assumption (1).

$A = \cup_{i=1}^n A_i$ denotes the global alphabet and $P_i : A^* \rightarrow A_i^*$ the projections to the local alphabets. The concept of inverse projection: $P_i^{-1} : \text{Pwr}(A_i^*) \rightarrow \text{Pwr}(A^*)$ is also used. Recall the basic definition of controllability from [10].

Definition 3.1 (controllability): A partial language $K \subseteq L$ is said to be controllable with respect to L and A_u , if

$$K^2 A_u \cap L^2 \subseteq K^2$$

Denote the global plant and specification languages by K and L , respectively. In our setting, L is decomposable into local plant languages: $L = L_1 \parallel \dots \parallel L_n$ (note

that the L_i may have different alphabets). In most of the works on this topic K is similarly decomposable into local specification languages and $K \subseteq L$. The general case is when this condition is not satisfied and moreover K may not be included in L . This case has been studied in [2], where the assumption that all shared event are controllable is used. The condition of G -observability was needed for local computation of the supremal controllable sublanguage. In this paper we consider a stronger condition of G -controllability.

Instead of local specifications, languages $K_i := K \cap P_i^{-1}(L_i)$ are considered. These will play the role of local components of specification languages, although their alphabet is the global alphabet A . Still we will see some analogy. They can be viewed as local over-approximations of $K \cap L$, because clearly $K \cap L = \cap_{i=1}^n K_i$. However, it turns out that it is not possible to compute simply the supremal controllable sublanguages and then take their intersection. That expression yields a language which in general is too small. Instead, another approach is proposed in [2] using the newly introduced concept of partial controllability.

Definition 3.2 (partial controllability): A partial language $K' \subseteq K \subseteq L$ is said to be partially controllable with respect to A' , A (with $A' \subseteq A$), K , and L , if

- (i) K' is controllable with respect to A' and L
- (ii) K' is controllable with respect to A and K .

Remark 3.1: Since controllability concerns only second (prefix-closed) components of partial languages, an order relation on partial languages induced by second components only is used: we write $K \subseteq L$ iff $K^2 \subseteq L^2$.

It is shown in [2] that the supremal partially controllable sublanguage of K with respect to A' , A (with $A' \subseteq A$), K , and L , denoted by $\text{sup PC}(K, A', A, L)$ exists. We use the following notation for the supremal controllable sublanguage of K with respect to L and A_u : $\text{sup C}(K, A_u, L) = K /_{A_u}^{SC} L$ (see appendix for the corresponding coinductive definition). According to [2] $\text{sup PC}(K, A', A, L)$ can be computed from the formula (using our coalgebraic notation for supremal controllable sublanguage):

$$\text{sup PC}(K, A', A, L) = [K /_{A'}^{SC} L] /_A^{SC} K$$

Let us introduce the following concept, called G -controllability, where the input derivatives of K_i and $P_i^{-1}(L_i)$ are involved (see Appendix A.2)

Definition 3.3 (G -controllability): A specification language K is said to be G -controllable if $\forall i \in \{1, \dots, n\}$ and $\forall s \in K_i^2 = K^2 \cap P_i^{-1}(L_i)^2$: $(K_i)_s \cap A_u^*$ is controllable with respect to $(P_i^{-1}(L_i))_s \cap A_u^*$ and A_{iu} .

The notion of G -controllability is sufficient for the computation of the global supremal controllable sublanguages without building (and manipulating with)

the global plant. Sufficient conditions for modular control synthesis to equal global control synthesis in the case of complete observations and of indecomposable specifications are formulated below.

Theorem 3.2: Let K be G -controllable and $\forall i \neq j \in \{1, \dots, n\}$ we have $A_{iu} \cap A_j = A_i \cap A_{ju}$. Then

$$\bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i = (K \cap L) /_{A_u}^{SC} L.$$

Proof: Notice that the expression for the supremal partially controllable sublanguage, consisting of two iterations of supremal controllable sublanguages, is used within the above formula. The coinductive proof principle will be used, i.e. it is sufficient to show that

$$R = \{ \langle (K \cap L) /_{A_u}^{SC} L, \bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \rangle; K, L \in \mathcal{L} \}$$

is a bisimulation relation, from which the equality follows by coinduction (see Appendix A.2).

(i) This is obvious from the coinductive definition of supremal controllable sublanguages.

(ii) If for $a \in A$ we have $\bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \xrightarrow{a}$ then $\forall i \in \mathbb{Z}_n : [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \xrightarrow{a}$. Thus, we have for $a \in A$ that $\forall i \in \mathbb{Z}_n : K_i \xrightarrow{a}$, $[K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] \xrightarrow{a}$ and $\forall u \in A_u^* : (K_i)_a \xrightarrow{u} \Rightarrow [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)]_a \xrightarrow{u}$. From the coinductive characterization of the inner supremal controllable sublanguage we further obtain that if $(K_i)_a \xrightarrow{u}$ then $P_i^{-1}(L_i)_a \xrightarrow{u}$ and $\forall v \in A_{iu}^* : P_i^{-1}(L_i)_{au} \xrightarrow{v} \Rightarrow (K_i)_{au} \xrightarrow{v}$.

We must show that $(K \cap L) /_{A_u}^{SC} L \xrightarrow{a}$, which according to the coinductive definition of the supremal controllable sublanguage means that $(K \cap L) \xrightarrow{a}$, $L \xrightarrow{a}$, and $\forall u \in A_u^* : L_a \xrightarrow{u} \Rightarrow (K \cap L)_a \xrightarrow{u}$. First of all, $(K \cap L) \xrightarrow{a}$ immediately follows from $K_i \xrightarrow{a}$, because $K \cap L = K \cap \bigcap_{i=1}^n P_i^{-1}(L_i) = \bigcap_{i=1}^n [K \cap P_i^{-1}(L_i)] = \bigcap_{i=1}^n K_i$. Now let $u \in A_u^*$ such that $L_a \xrightarrow{u}$. We can write $u = u_1 \dots u_N$, where $\forall i \in \mathbb{Z}_N$ there exists $k_i \in \mathbb{Z}_n$ such that $u_i \in A_{k_i u} \subseteq A_u$. Note that $\forall i \in \mathbb{Z}_n$ we have $L^2 \subseteq P_{k_i}^{-1}(L_{k_i})^2$. For $k_1 \in \mathbb{Z}_n$ we obtain from $L_a \xrightarrow{u_1}$ that in particular $P_{k_1}^{-1}(L_{k_1})_a \xrightarrow{u_1}$. From our assumptions we have in particular that $[K_{k_1} /_{A_{k_1 u}}^{SC} P_{k_1}^{-1}(L_{k_1})] \xrightarrow{a}$, i.e. according to the coinductive definition of the local (inner) supremal controllable sublanguage $P_{k_1}^{-1}(L_{k_1})_a \xrightarrow{u_1}$ implies that $(K_{k_1})_a \xrightarrow{u_1}$. Since by definition $K_{k_1} \subseteq K$ we obtain that $(K \cap L)_a \xrightarrow{u_1}$ as well. Now from $K_{k_2} = K \cap P_{k_2}^{-1}(L_{k_2})$ and $P_{k_2}^{-1}(L_{k_2})_a \xrightarrow{u_1}$ we have $(K_{k_2})_a \xrightarrow{u_1}$. Since $u_1 \in A_u^*$ we obtain according to the coinductive definition of the global (outer) supremal controllable sublanguage that $(K_{k_2})_a \xrightarrow{u_1}$ implies that $[K_{k_2} /_{A_{k_2 u}}^{SC} P_{k_2}^{-1}(L_{k_2})]_a \xrightarrow{u_1}$. Now for k_2 we obtain from $L_a \xrightarrow{u_1 u_2}$ that $P_{k_2}^{-1}(L_{k_2})_{au_1} \xrightarrow{u_2}$, i.e. from $[K_{k_2} /_{A_{k_2 u}}^{SC} P_{k_2}^{-1}(L_{k_2})]_a \xrightarrow{u_1}$ and $u_2 \in A_{k_2 u}$ it follows that

$(K_{k_2})_{au_1} \xrightarrow{u_2}$. In particular we have that $(K \cap L)_{au_1} \xrightarrow{u_2}$. We continue inductively this way along the string u . We obtain finally that $(K_{i_n})_{au_1 \dots u_{n-1}} \xrightarrow{u_n}$. Since $K_{i_n} = K \cap P_{i_n}^{-1}(L_{i_n})$ we have finally that $(K \cap L)_a \xrightarrow{u}$, because $L_a \xrightarrow{u}$.

(iii) If $(K \cap L) /_{A_u}^{SC} L \xrightarrow{a}$ for $a \in A$ then it follows from the coinductive definition of the supremal controllable sublanguage that $(K \cap L) \xrightarrow{a}$, $L \xrightarrow{a}$, and $\forall u \in A_u^* : L_a \xrightarrow{u} \Rightarrow (K \cap L)_a \xrightarrow{u}$. We need to show that $\bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \xrightarrow{a}$, i.e. $\forall i \in \mathbb{Z}_n : [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \xrightarrow{a}$. According to the coinductive characterization of the supremal partially controllable sublanguage, which is also used in (ii) above we need to show that $\forall i \in \mathbb{Z}_n : K_i \xrightarrow{a}$, $[K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] \xrightarrow{a}$ and $\forall u \in A_u^* : (K_i)_a \xrightarrow{u} \Rightarrow [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)]_a \xrightarrow{u}$. It follows further from the coinductive characterization of the inner supremal controllable sublanguage that if $(K_i)_a \xrightarrow{u}$ then $P_i^{-1}(L_i)_a \xrightarrow{u}$ and $\forall v \in A_{iu}^* : P_i^{-1}(L_i)_{au} \xrightarrow{v} \Rightarrow (K_i)_{au} \xrightarrow{v}$.

The first claim is obvious from $(K \cap L) = \bigcap_{i=1}^n K_i$: $(K \cap L) \xrightarrow{a}$ implies that $\forall i \in \mathbb{Z}_n : (K_i) \xrightarrow{a}$. The second claim: $[K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] \xrightarrow{a}$ is a special case of the claim $\forall u \in A_u^* : (K_i)_a \xrightarrow{u} \Rightarrow [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)]_a \xrightarrow{u}$ for $u = \varepsilon$. Let $(K_i)_a \xrightarrow{u}$ for a $u \in A_u^*$. We need to show that $[K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)]_a \xrightarrow{u}$, which according to the coinductive definition of the inner supremal controllable sublanguage means that $P_i^{-1}(L_i)_a \xrightarrow{u}$, $(K_i)_a \xrightarrow{u}$ (trivially satisfied) and $\forall v \in A_{iu}^* : P_i^{-1}(L_i)_{au} \xrightarrow{v} \Rightarrow (K_i)_{au} \xrightarrow{v}$. $P_i^{-1}(L_i)_a \xrightarrow{u}$ is obvious from $(K_i)_a \xrightarrow{u}$, because $\forall i : K_i \subseteq P_i^{-1}(L_i)$. The rest follows from G -controllability: if $P_i^{-1}(L_i)_{au} \xrightarrow{v}$ for some $u \in A_u^*$ and $v \in A_{iu}^*$, then $auv \in P_i^{-1}(L_i)^2$, i.e. $uv \in P_i^{-1}(L_i)_a^2 \cap A_u^*$. Moreover, $(K_i)_a \xrightarrow{u}$, i.e. $au \in (K_i)^2$ implies that $u \in (K_i)_a^2 \cap A_u^*$. Thus, according to G -controllability $uv \in (K_i)_a^2$, which is equivalent to $(K_i)_{au} \xrightarrow{v}$, which was to be shown. ■

Remark 3.3: (i) It is obvious that our condition of G -controllability is stronger than the condition of G -observability introduced in [2]. On the other hand, G -controllability cannot be replaced by G -observability. This works only in the special setting of [2], where all shared events are controllable.

(ii) It is obvious from the above proof that for (i) and (ii) of the proof of Theorem 3.2, G -controllability is not needed, i.e. under very general conditions we always have the inclusion:

$$\bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \subseteq (K \cap L) /_{A_u}^{SC} L.$$

(iii) Note finally that the result of Theorem 3.2 is not effective. Indeed, G -controllability is of limited interest, because it implies that all K_i are partially controllable, and thus $K \cap L$ is controllable with respect to L and A_u . In this case the terms on both sides of the claimed equality are equal to $K \cap L$ itself. We have only an existential result that can be viewed as a low complexity test for controllability. Effective results are

in the sequel of this paper. The reason why G -controllability of K implies that all K_i are partially controllable is that for $u = \varepsilon \in A_u^*$, thus G -controllability also means that (i) of partial controllability for $K = K_i$ holds true, while (ii) is trivial.

As an alternative way of ensuring the other inclusion we can use a structural condition similar to the mutual controllability of [12]. Following the same idea as in the case of decomposable specification [4] we introduce the following concept:

Definition 3.4 (Strong global mutual controllability): Local plant (partial) languages L_i , $i \in \mathbb{Z}_n$ are called globally mutually controllable if for any $i \neq j \in \mathbb{Z}_n$ we have

$$P_j^{-1}(L_j^2)A_u \cap P_i^{-1}(L_i^2) \subseteq P_j^{-1}(L_j^2).$$

Although the condition concerns languages $P_i^{-1}(L_i)$ over global alphabet, these are easily derived from the local plant languages L_i and we still avoid building the representation of the whole plant. We notice that the recognizers of $P_i^{-1}(L_i)$ are easily obtained from the recognizers of L_i by simply adding to all states the self-loops of events that are not in A_i . Unlike a typical situation in supervisory control, strong global mutual controllability is a symmetric notion of controllability, where it is not true that one language is a sublanguage of the other. Now we can formulate our main theorem. Sufficient structural conditions for modular control synthesis to equal global control synthesis in the case of complete observations and of indecomposable specifications are formulated below.

Theorem 3.4: Let L_i , $i \in \mathbb{Z}_n$ be strongly globally mutually controllable and $\forall i \neq j \in \{1, \dots, n\}$ we have $A_{iu} \cap A_j = A_i \cap A_{ju}$. Then

$$\bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i = (K \cap L) /_{A_u}^{SC} L.$$

Proof: The coinductive proof principle will be used, i.e. it is sufficient to show that R from the proof of Theorem 3.2 is again a bisimulation relation in this setting.

(i) and (ii) are the same as in the preceding theorem, because the assumption of G -controllability is not used. It suffices to show

(iii) Let $(K \cap L) /_{A_u}^{SC} L \xrightarrow{a}$ for $a \in A$. It follows from the coinductive definition of the supremal controllable sublanguage that $(K \cap L) \xrightarrow{a}$, $L \xrightarrow{a}$, and $\forall u \in A_u^* : L_a \xrightarrow{u} \Rightarrow (K \cap L)_a \xrightarrow{u}$. We need to show that $\bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \xrightarrow{a}$, i.e. that $\forall i \in \mathbb{Z}_n : [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i \xrightarrow{a}$. According to (iii) of the proof of the previous theorem, this amounts to show the part, where G -controllability is used. Let us show that $\forall u \in A_u^*$ and $\forall v \in A_{iu}^* : P_i^{-1}(L_i)_{au} \xrightarrow{v} \Rightarrow (K_i)_{au} \xrightarrow{v}$. For this implication global mutual controllability is used.

If $P_i^{-1}(L_i)_{au} \xrightarrow{v}$ for some $u \in A_u^*$ and $v \in A_{iu}^*$, then $auv \in P_i^{-1}(L_i)^2$. Since $uv \in A_u^*$ we obtain that $uv = v_1 \dots v_k$ for some $k \in \mathbb{N}$, where $v_i \in A_u$, $i \in \mathbb{Z}_k$. Now we proceed by induction along the string v . According to global mutual controllability we obtain $av_1 \in P_j^{-1}L_j^2(A_u) \cap P_i^{-1}L_i^2 \subseteq P_j^{-1}L_j^2$. Thus, in both cases $av_1 \in L^2 = \bigcap_{i=1}^n P_i^{-1}(L_i^2)$, which is equivalent to $L_a \xrightarrow{v_1}$. Similar argument is made for any v_l , $l \in \mathbb{Z}_k$. Thus, we obtain after an inductive application of the same argument that $auv = av_1 \dots v_k \in L^2$, which is equivalent to $L_{au} \xrightarrow{v}$. Notice that $uv \in A_u^*$. A direct application of the assumption that $(K \cap L) /_{A_u}^{SC} L \xrightarrow{a}$ now yields $(K \cap L)_a \xrightarrow{uv}$, which means that $auv \in (K \cap L)^2 = \bigcap_{i=1}^n K \cap P_i^{-1}(L_i^2)$, i.e. $\forall i \in \mathbb{Z}_n : \text{we have } (K_i)_{au} = (K \cap P_i^{-1}(L_i^2))_{au} \xrightarrow{v}$, which was to be shown. ■

The last theorem provides a structural condition under which supremal controllable sublanguages can be computed without having to build the global plant. Moreover as a structural condition, it does not depend on a particular specification, which is very important for indecomposable specifications.

The condition of strong global mutual controllability is too restrictive, thus we adopt a weaker condition of global mutual controllability. Nevertheless, it turns out that we need moreover a condition from [2]. We obtain a combined, more elaborate, approach and use the condition of local consistency.

Definition 3.5 (Global mutual controllability): Local plant (partial) languages L_i , $i \in \mathbb{Z}_n$ are called globally mutually controllable if for any $i \neq j \in \mathbb{Z}_n$ we have

$$P_j^{-1}(L_j^2)(A_{ju} \cap A_i) \cap P_i^{-1}(L_i^2) \subseteq P_j^{-1}(L_j^2).$$

Definition 3.6 (Local consistency): A global specification K is said to be locally consistent with respect to A_u and L_i , $i \in \mathbb{Z}_n$ if for any $i \in \mathbb{Z}_n$ we have: $\forall s \in K_i^2$ and $\forall u \in A_u^*$ such that $su \in K_i^2$ and $\forall v \in A_{iu}^* : sP_i(u)v \in K_i^2 \Rightarrow suv \in K_i^2$

Our main theorem follows. It provides sufficient structural conditions for modular control synthesis to equal global control synthesis in the case of complete observations and of indecomposable specifications.

Theorem 3.5: Let L_i , $i \in \mathbb{Z}_n$ be globally mutually controllable and $\forall i \neq j \in \{1, \dots, n\}$ we have $A_{iu} \cap A_j = A_i \cap A_{ju}$. Then

$$\bigcap_{i=1}^n [K_i /_{A_{iu}}^{SC} P_i^{-1}(L_i)] /_{A_u}^{SC} K_i = (K \cap L) /_{A_u}^{SC} L.$$

Proof: The coinductive proof principle will be used, i.e. it is sufficient to show that R from the proof of Theorem 3.2 is again a bisimulation relation in this setting.

(i) and (ii) are the same as in the preceding theorem, because

the assumption of G -controllability is not used. It suffices to show

(iii) Let $(K \cap L)/_{A_u}^{SC} L \xrightarrow{a}$ for $a \in A$. It follows from the coinductive definition of the supremal controllable sublanguage that $(K \cap L) \xrightarrow{a}$, $L \xrightarrow{a}$, and $\forall u \in A_u^* : L_a \xrightarrow{u} \Rightarrow (K \cap L)_a \xrightarrow{u}$. We need to show that $\bigcap_{i=1}^n [K_i/_{A_{iu}}^{SC} P_i^{-1}(L_i)]/_{A_u}^{SC} K_i \xrightarrow{a}$, i.e. that $\forall i \in \mathbb{Z}_n : [K_i/_{A_{iu}}^{SC} P_i^{-1}(L_i)]/_{A_u}^{SC} K_i \xrightarrow{a}$. According to (iii) of the proof of the previous theorem, this amounts to show the part, where G -controllability is used. Let us show that $\forall u \in A_u^*$ and $\forall v \in A_{iu}^* : P_i^{-1}(L_i)_{au} \xrightarrow{v} \Rightarrow (K_i)_{au} \xrightarrow{v}$. For this implication global mutual controllability and local consistency are used. If $P_i^{-1}(L_i)_{au} \xrightarrow{v}$ for some $u \in A_u^*$ and $v \in A_{iu}^*$, then $auv \in P_i^{-1}(L_i)^2$, i.e. $aP_i(u)v \in P_i^{-1}(L_i)^2$ as well. Since $P_i(u)v \in A_{iu}^*$ we obtain that $P_i(u)v = v_1 \dots v_k$ for some $k \in \mathbb{N}$, where $v_i \in A_{iu}$, $i \in \mathbb{Z}_k$. Now we proceed by induction along the string v . We know that $v \in A_{iu}$. For any $j \in \mathbb{Z}_n : j \neq i$ we have according to our assumption that $A_{iu} \cap A_j = A_i \cap A_{ju}$ two possibilities: either $v_1 \in A_{ju}$ and then $v_1 \in A_{iu} \cap A_j$ or $v_1 \notin A_j$. The case $v_1 \notin A_j$ is easy, because then $P_j(v_1) = \varepsilon$, i.e. $P_j(av_1) = P_j(a)$. Since $a \in L^2$ we have also $a \in P_j^{-1}(L_j)^2$, thus $av_1 \in P_j^{-1}(L_j)^2$ as well.

If $v_1 \in A_{iu} \cap A_j$, then according to global mutual controllability we obtain $av_1 \in P_j^{-1}L_j^2(A_{ju} \cap A_i) \cap P_i^{-1}L_i^2 \subseteq P_j^{-1}L_j^2$. Thus, in both cases $av_1 \in L^2 = \bigcap_{i=1}^n P_i^{-1}(L_i^2)$, which is equivalent to $L_a \xrightarrow{v_1}$. Similar construction is made for any v_l , $l \in \mathbb{Z}_k$, where cases $v_l \in A_{ju}$ and $v_l \notin A_j$ are distinguished. Thus, we obtain after an inductive application of the same argument along the string $P_i(u)v \in A_{iu}^*$ that $aP_i(u)v = auv_1 \dots v_k \in L^2$, which is equivalent to $L_a \xrightarrow{P_i(u)v}$. Notice that in particular $P_i(u)v \in A_{iu}^*$. A direct application of the assumption that $(K \cap L)/_{A_u}^{SC} L \xrightarrow{a}$ now yields $(K \cap L)_a \xrightarrow{P_i(u)v}$, which means that $aP_i(u)v \in (K \cap L)^2 = \bigcap_{i=1}^n K \cap P_i^{-1}(L_i^2)$, i.e. $aP_i(u)v \in K_i^2$. This implies using local consistency with $s = a$ that $su v \in K_i^2$. Therefore $\forall i \in \mathbb{Z}_n : \text{we have } (K_i)_{au} = (K \cap P_i^{-1}(L_i^2))_{au} \xrightarrow{v}$, which was to be shown. ■

IV. CONCLUSION

New methods for modular computation of supremal controllable sublanguages of indecomposable specification languages have been presented. Although the proofs of the main results seem at the first time technically very complicated, their principle is very simple and a small number of the same arguments is repeatedly used. All the sufficient conditions we have presented can be checked easier than their counterparts for global systems. The structural condition of strong global mutual controllability does not depend on a particular specification, which is very important because general indecomposable specifications are studied. However, it is too strong, and therefore our main result uses a weaker version that we call simply global mutual controllability.

REFERENCES

- [1] S.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, Dordrecht 1999.
- [2] B. Gaudin and H. Marchand. Modular Supervisory Control of a Class of Concurrent Discrete Event Systems. Proceedings *WODES'04*, Workshop on Discrete-Event Systems, pp. 181-186, Reims, September 22-24, 2004.
- [3] J. Komenda and J.H. van Schuppen. Control of Discrete-Event Systems with Partial Observations Using Coalgebra and Coinduction, *Discrete Event Dynamic Systems: Theory and Applications* 15(3):257-315, 2005.
- [4] J. Komenda and J.H. van Schuppen. Supremal Normal Sublanguages of Large Distributed Discrete-Event Systems. Proceedings *WODES'04*, Workshop on Discrete-Event Systems, pp. 73-78, Reims, September 22-24, 2004.
- [5] F. Lin and W.M. Wonham. Decentralized Supervisory Control of Discrete-Event Systems, *Information Sciences* 44:199-224, 1988.
- [6] R. Milner. *Communication and Concurrency*. Prentice Hall International Series in Computer Science. Prentice Hall International, New York, 1989.
- [7] K. Rohloff and S. Lafortune. On the Computational Complexity of the Verification of Modular Discrete-Event Systems. In *Proc. 41 st IEEE Conference on Decision and Control, Las Vegas, Nevada, USA, December 2002*.
- [8] J.J.M.M. Rutten. Coalgebra, Concurrency, and Control. *Research Report CWI, SEN-R9921*, Amsterdam, November 1999. Available also at <http://www.cwi.nl/~janr>.
- [9] J.J.M.M. Rutten. Universal Coalgebra: A Theory of Systems. *Theoretical Computer Science* 249:3-80, 2000.
- [10] P.J. Ramadge and W.M. Wonham. The Control of Discrete-Event Systems. *Proc. IEEE*, 77:81-98, 1989.
- [11] Y. Willner and M. Heymann. Supervisory Control of Concurrent Discrete-Event Systems. *International Journal of Control*, 54:1143-1166, 1991.
- [12] K.C. Wong and S. Lee. Structural Decentralized Control of Concurrent Discrete-Event Systems. *European Journal of Control*, 8:477-491, 2002.
- [13] K.C. Wong and W.M. Wonham. Modular Control and Coordination of Discrete-Event Systems. *Discrete Event Dynamical Systems: Theory and Applications*, 8:247-297, 1998.
- [14] W.M. Wonham and P.J. Ramadge. Modular Supervisory Control of Discrete-Event Processes, *Mathematics of Control, Signal and Systems*, 1:13-30, 1988.

APPENDIX

Coalgebras are categorical duals of algebras (the corresponding functor operates from a given set rather than to a given set). The basic introduction to the theory of universal coalgebra is developed in analogy with the corresponding theory of universal algebra in [9]. The concept of final coalgebras enables definitions and proofs by coinduction.

A. Partial automata

In this section partial automata as generators of DES are formulated coalgebraically as in (Rutten 1999). Final coalgebra of partial automata, i.e. a partial automaton of partial languages is then recalled. Let A be the set of events. The empty string will be denoted by ε . Denote by $1 = \{\emptyset\}$ the one element set and by $2 = \{0, 1\}$ the set of Booleans. A partial automaton is a pair $S = (S, \langle o, t \rangle)$, where S is a set of states, and a pair of functions $\langle o, t \rangle : S \rightarrow 2 \times (1 + S)^A$, consists of an output function $o : S \rightarrow 2$ and a transition function $t : S \rightarrow (1 + S)^A$. The output function o indicates whether a state $s \in S$ is accepting (or terminating) : $o(s) = 1$, denoted also by $s \downarrow$, or not: $o(s) = 0$, denoted by $s \uparrow$. The transition function t associates to each state s in S

a function $t(s) : A \rightarrow (1 + S)$. The set $1 + S$ is the disjoint union of S and 1 . The meaning of the state transition function is that $t(s)(a) = \emptyset$ iff $t(s)(a)$ is undefined, which means that there is no a -transition from the state $s \in S$. $t(s)(a) \in S$ means that the a -transition from s is possible and we define in this case $t(s)(a) = s_a$, which is denoted mostly by $s \xrightarrow{a} s_a$.

A *bisimulation* between two partial automata $S = (S, \langle o, t \rangle)$ and $S' = (S', \langle o', t' \rangle)$ is a relation $R \subseteq S \times S'$ such that: if $\langle s, s' \rangle \in R$ then

- (i) $o(s) = o(s')$, i.e. $s \downarrow$ iff $s' \downarrow$
- (ii) $\forall a \in A : s \xrightarrow{a} \Rightarrow (s' \xrightarrow{a} \text{ and } \langle s_a, s'_a \rangle \in R)$,
- (iii) $\forall a \in A : s' \xrightarrow{a} \Rightarrow (s \xrightarrow{a} \text{ and } \langle s_a, s'_a \rangle \in R)$.

We write $s \sim s'$ whenever there exists a bisimulation R with $\langle s, s' \rangle \in R$. This relation is the union of all bisimulations, i.e. the greatest bisimulation also called bisimilarity.

B. Final automaton of partial languages

Below we define the partial automaton of partial languages over an alphabet (input set) A , denoted by $\mathcal{L} = (\mathcal{L}, \langle o_{\mathcal{L}}, t_{\mathcal{L}} \rangle)$. More formally,

$$\mathcal{L} = \{(V, W) \mid V \subseteq W \subseteq A^*, W \neq \emptyset, \text{ and } W \text{ is prefix-closed}\}.$$

The transition function $t_{\mathcal{L}} : \mathcal{L} \rightarrow (1 + \mathcal{L})^A$ is defined using input derivatives. Recall that for any partial language $L = (L^1, L^2) \in \mathcal{L}$, $L_a = (L_a^1, L_a^2)$, where $L_a^i = \{w \in A^* \mid aw \in L^i\}$, $i = 1, 2$. If $a \notin L^2$ then L_a is undefined. Given any $L = (L^1, L^2) \in \mathcal{L}$, the partial automaton structure of \mathcal{L} is given by:

$$o_{\mathcal{L}}(L) = \begin{cases} 1 & \text{if } \varepsilon \in L^1 \\ 0 & \text{if } \varepsilon \notin L^1 \end{cases}$$

$$t_{\mathcal{L}}(L)(a) = \begin{cases} L_a & \text{if } L_a \text{ is defined} \\ \emptyset & \text{otherwise} \end{cases}$$

Notice that if L_a is defined, then $L_a^1 \subseteq L_a^2$, $L_a^2 \neq \emptyset$, and L_a^2 is prefix-closed. The following notational conventions will be used: $L \downarrow$ iff $\varepsilon \in L^1$, and $L \xrightarrow{w} L_w$ iff L_w is defined iff $w \in L^2$.

Recall from (Rutten 1999) that $\mathcal{L} = (\mathcal{L}, \langle o_{\mathcal{L}}, t_{\mathcal{L}} \rangle)$ is final among all partial automata: for any partial automaton $S = (S, \langle o, t \rangle)$ there exists a unique homomorphism $l : S \rightarrow \mathcal{L}$. Another characterization of finality of \mathcal{L} is that it satisfies the principle of coinduction: for all K and L in \mathcal{L} , if $K \sim L$ then $K = L$. Recall that the unique homomorphism l given by finality of \mathcal{L} maps a state $s \in S$ to $l(s) = (L_s^1, L_s^2) = (\{w \in A^* \mid s \xrightarrow{w} \text{ and } s_w \downarrow\}, \{w \in A^* \mid s \xrightarrow{w}\}) \in \mathcal{L}$.

C. Coinductive definitions

Coinduction is a concept that is dual to induction. Coinduction in its full generality must be put into a general framework of universal coalgebra that uses the category theory. In order to make the paper more accessible to a reader not very familiar with category theory we have preferred to introduce the coinduction only in its special

form: on final coalgebra of partial languages. It is the same as with mathematical induction that is by many people understood only on the initial algebra of natural numbers (with the structure given by successor operation: $\forall n \in \mathbb{N} : succ(n) = n + 1$), where definitions of functions by induction correspond to giving initial values and successors on functions, hence yielding recursive formulas, and proofs by induction correspond to the famous two-step procedure, which amounts to verify that a relation is a congruence relation with respect to the successor operation. Similarly, a definition by coinduction amounts to defining the corresponding structure, here output and derivatives on operations to be defined, and a proof by coinduction consists in verifying the conditions of a bisimulation relation.

Recall from [8] the following coinductive definitions of the synchronous and the supervised products with complete observations. For the synchronous product we assume that K is defined over the alphabet A_1 and L over A_2 . Then the synchronous product $K \parallel L$ is a language over $A_1 \cup A_2$ with the following coinductive definition:

Definition (Synchronous product)

$$(K \parallel L)_a = \begin{cases} K_a \parallel L_a & \text{if } a \in A_1 \cap A_2 \\ K_a \parallel L & \text{if } a \in A_1 \setminus A_2 \\ K \parallel L_a & \text{if } a \in A_2 \setminus A_1 \end{cases}$$

and $(K \parallel L) \downarrow$ iff $K \downarrow$ and $L \downarrow$.

In the definitions below $A_u \subseteq A$ stands for the subset of uncontrollable events. Now we recall from [3] the following binary operation on partial languages:

Definition (Supremal controllable sublanguage) Define the following binary operation on (partial) languages for all $K, L \in \mathcal{L}$ and $\forall a \in A$:

$$(K /_{A_{uc}}^S L)_a = \begin{cases} K_a /_C^S L_a & \text{if } K \xrightarrow{a} \text{ and } L \xrightarrow{a} \\ & \text{and if } \forall u \in A_u^* : \\ & L_a \xrightarrow{u} \Rightarrow K_a \xrightarrow{u} \\ \emptyset & \text{otherwise} \end{cases}$$

and $(K /_C^S L) \downarrow$ iff $L \downarrow$.

We have shown in [3] that for a partial order that considers only second (prefix-closed) components of the languages involved:

Theorem $(K /_C^S L) = \sup(\underline{C}(K, L)) = \sup\{M \subseteq K : M \text{ is controllable with respect to } L \text{ and } A_{uc}\}$, i.e. $K /_C^S L$ equals the supremal controllable sublanguage of K .