Modular and Decentralized Supervisory Control of Concurrent Discrete Event Systems Using Reduced System Models.

Klaus Schmidt, Hervé Marchand, Benoît Gaudin

To cite this version:


HAL Id: inria-00425106
https://hal.inria.fr/inria-00425106
Submitted on 20 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Modular and Decentralized Supervisory Control of Concurrent Discrete Event Systems Using Reduced System Models

Klaus Schmidt
Universität Erlangen-Nürnberg
Lehrstuhl für Regelungstechnik
Cauerstraße 7 - 91058 Erlangen, Germany
klaus.schmidt@rt.eei.uni-erlangen.de

Hervé Marchand
IRISA-INRIA Rennes
Campus univ de Rennes 1
35042 Rennes, France
herve.marchand@irisa.fr

Benoît Gaudin
FOKUS Fraunhofer
Kaiserin Augusta Allee 31
10589 Berlin, Germany
bg@fokus.fraunhofer.de

Abstract—This work investigates the supervisor synthesis for concurrent systems based on reduced system models with the intention of complexity reduction. It is assumed that the expected behavior (specification) is given on a subset of the system alphabet, and the system behavior is reduced to this alphabet. Supervisors are computed for each reduced subsystem employing the modular approach in [5] and the decentralized approach in [8]. Depending on the chosen architecture, we provide sufficient conditions for the consistent implementation of the reduced supervisors for the original system.

Keywords—Concurrent discrete event systems, hierarchical control, modular and decentralized architecture.

I. INTRODUCTION

The main issue in supervisor synthesis for discrete event systems (DES) is the state-space explosion for large-scale systems. Addressing this problem, recent approaches study hierarchical, decentralized and modular methods to reduce the complexity of supervisor synthesis algorithms.

In hierarchical architectures [17], [3], [7], [14], [9], [13], controller synthesis is based on a plant abstraction (high-level model), which is supposed to be less complex than the original plant model (low-level model).

The structure of concurrent systems (systems modeled by several components) is exploited for decentralized and modular control. In most of the decentralized architectures [15], [10], [1], [11], [6], [8], the methodology is characterized by the fact that the specification (i.e. the expected behavior) can be decomposed according to the structure of the plant. In that case, local modular supervisors operating each concurrent system component individually are implemented, and necessary and sufficient conditions under which the behavior of the controlled plant corresponds to the supremal one are given. In contrast, the authors of [4], [5] consider a modular architecture. The specification does not need to be separable (but locally consistent and prefix-closed, which is not the case for most of the previously mentioned works). Modular supervisors can be computed based on the specification and abstractions of the subsystems so that they solve the supervisory control problem without having to build the whole system.

In this paper, we elaborate two approaches for concurrent systems that both avoid the computation of the overall system and are based on a reduced system model. We assume that the specification is given on a subset of the system alphabet and the behavior of the concurrent systems is reduced to this alphabet. Supervisors are synthesized for the reduced system models using the modular approach in [5] and the decentralized approach in [8]. We provide sufficient conditions for the consistent implementation of the reduced supervisors for the original system.

The outline of the paper is as follows. After providing basic definitions in supervisory control in Section II, we present the setting of the paper in Section III. Section IV and V discuss modular and structural decentralized control for reduced system models, respectively. Conclusions are given in Section VI.

II. PRELIMINARIES

We recall basics from supervisory control theory [18], [2].

For a finite alphabet , the set of all finite strings over is denoted . Write for the concatenation of two strings and . We write when is a prefix of , i.e. if there exists a string with . The empty string is denoted , i.e. . A language over is a subset . The prefix closure of is defined by if and only if for all in the language . A language is prefix closed if . Let , then if is nonblocking w.r.t. if .

The natural projection for the (not necessarily disjoint) union is for the set of events . The set-valued inverse of is denoted . The synchronous product of and is denoted . A finite automaton is a tuple , with the finite set of states , the finite alphabet of events , the partial transition function , the initial
state $x_0 \in X$; and the set of marked states $X_m \subseteq X$. We write $\delta(x, \sigma)$ if $\delta$ is defined at $(x, \sigma)$. In order to extend $\delta$ to a partial function on $X \times \Sigma^*$, recursively let $\delta(x, \epsilon) := x$ and $\delta(x, s\sigma) := \delta(\delta(x, s), \sigma)$, whenever both $x' = \delta(x, s)$ and $\delta(x', \sigma)$. A language $L(G)$ is the set of strings accepted by $G$. A language $L$ is prefix-closed, if $s \in L \Rightarrow \sigma \in L$ for each string $s$ and the set $\Sigma$.

In a supervisory control context, we write $\Sigma = \Sigma_c \cup \Sigma_{uc}$, where $\Sigma_c \cap \Sigma_{uc} = \emptyset$, to distinguish controllable ($\Sigma_c$) and uncontrollable ($\Sigma_{uc}$) events. A control pattern is a set $\gamma$, $\Sigma_{uc} \subseteq \gamma \subseteq \Sigma$, and the set of all control patterns is denoted $\Gamma \subseteq 2^\Sigma$. A supervisor is a map $S: L(G) \rightarrow \Gamma$, where $\Gamma(s)$ represents the set of enabled events after the occurrence of string $s$; i.e., a supervisor can disable controllable events only. The behavior of the closed-loop system is characterized by $L(S/G)$ generated by $G$ under supervision $S$.

A language $H$ is said to be controllable w.r.t. $L(G)$ and $\Sigma_{uc}$ if there exists a supervisor $S$ such that $\overline{L} = L(S/G)$. The set of all languages that are controllable w.r.t. $L(G)$ is denoted $C(L(G))$ and can be characterized by $C(L(G)) = \{H \subseteq L(G) \; \exists S \in S(G) \}$. Furthermore, the set $C(L(G))$ is closed under arbitrary union. Hence, for every specification language $E$ there uniquely exists a supremal controllable sublanguage w.r.t. $L(G)$ and $\Sigma_{uc}$, which is formally defined as $\overline{E} \subseteq \Sigma_{uc}$, if $E$ is controllable w.r.t. $\Sigma_c$ and $L(G)$ and $\Sigma_{uc}$.

As a system model, we consider concurrent DES represented by finite automata $(G_1)_{1 \leq i \leq n}$ over the corresponding alphabets $\Sigma_i = \Sigma_{i,uc} \cup \Sigma_{i,c}$. Here, $\Sigma_{i,uc}$ and $\Sigma_{i,c}$ denote the uncontrollable and the controllable events, respectively. We assume that all subsystems are directly or indirectly connected to all other subsystems via events from the set $\Sigma_s := \bigcup_{i,k \neq i} (\Sigma_i \cap \Sigma_k)$ of shared events. The global set of shared events is thus given by $\Sigma_s = \bigcup_{i} \Sigma_{i,s}$. The overall system model is $G := ||G_i||$ over the alphabet $\Sigma := \bigcup_i \Sigma_i$. Moreover, we assume that the components that share an event agree on the control status of this event, i.e., $\forall i, k$, $\Sigma_{i,uc} \cap \Sigma_{k,uc} = \emptyset$. Under this hypothesis, we have that $\Sigma_{uc} = \bigcup_i \Sigma_{i,uc}$ and $\Sigma_s = \bigcup_i \Sigma_{i,s}$.

The main objective of this paper is to study control architectures which reduce the computational complexity of supervisor synthesis for a given specification $K \subseteq \Sigma^*$ by avoiding the computation of $\overline{G}$. To this end, we are interested in the case where the complexity of the specification $K$ is lower than that of the plant $G$. In the literature, there are different approaches tackling this problem.

An approach for the modular control of concurrent systems is proposed in [4], [5]. Modular supervisors are computed using abstractions of the decentralized subsystems and corresponding local specifications. The supremal partially controllable sublanguages of the local specifications solve the supervisory control problem if the specification $K$ is locally consistent and prefix-closed, and the languages of the subsystems are mutually controllable.

The method in [8] suggests structural decentralized control. It requires the specification $K$ to be separable, i.e., $K = \overline{\tilde{p}_i(K)}$, where $\overline{\tilde{p}_i}$ is the natural projection. If the languages of the subsystems are mutually controllable and shared event marking, then using nonblocking local controllers for the specifications $\overline{\tilde{p}_i(K)}$ is equivalent to the nonblocking overall supervisor.

This approach is supplemented with hierarchical control in [14]. Monolithic control is applied to a reduced (hierarchical) system model which is derived by projecting the behavior of the original model to the set of shared events $\Sigma_s$. However, this approach requires the computation of an overall reduced system model which is not always feasible.

Motivated by these considerations, we elaborate two methods that employ reduced concurrent system models, but avoid computing an overall reduced system model. To this end, we investigate the case where the specification $K \subseteq (\Sigma)^*$ for the supervisory control problem is given on a reduced alphabet $\Sigma \subseteq \Sigma$ with $\Sigma_s \subseteq \Sigma$. Hence, the reduction is based on projecting out events that occur in only one subsystem. With the reduced decentralized alphabets $\Sigma_i := \Sigma \cap \Sigma_i$ and the decentralized natural projections $\tilde{p}_i^{dec}: \Sigma_i \rightarrow \Sigma_i$, the decentralized reduced system models are $(G_i)_{1 \leq i \leq n}$, where $L(G_i) = \tilde{p}_i^{dec}(L(\tilde{G}_i))$ and $L_m(G_i) = \tilde{p}_i^{dec}(L_m(\tilde{G}_i))$. In the following sections, we utilize the approaches in [5] and [8] to design supervisors for the reduced system models. Based on these supervisors, we provide conditions for the decentralized supervisor implementation $\tilde{S}$ for the original systems. The first approach results in an estimation.

1Definitions of these notions are given in Section IV and V.
2This assumption is not restricted. If $\Sigma_s \neq \emptyset$, $K^c = \overline{K} || (\Sigma - \Sigma_s)^c \subseteq (\Sigma \cup \Sigma)^c$ fulfills the requirement.
of the supremal controllable sublanguage of a prefix-closed non-separable specification. The second method provides an estimation of the supremal controllable and nonblocking sublanguage of a not necessarily prefix-closed but separable specification. Figure 1 illustrates the control scheme.

![Modular/Decentralized Supervisor](image)

Fig. 1. Modular/Decentralized Architecture

IV. MODULAR CONTROL

According to [5], modular supervisors \( S_i^{-1} : \Sigma^* \rightarrow \Gamma_i \) with the set of control patterns \( \Gamma_i := \{ \gamma \subseteq \Sigma|\Sigma_{uc} \subseteq \gamma \} \) are computed for the abstractions \( G_i^{-1} \) of the reduced system models and the local specifications \( K_i^{-1} := K \cap L(G_i^{-1}) \), where \( L(G_i^{-1}) = p_i^{-1}(L(G_i)) \subseteq \Sigma^* \), with the natural projection \( p_i : \Sigma^* \rightarrow \Sigma_i^* \). The main result of [5] is based on the following definitions.

**Definition 4.1:** \( G_i \) and \( G_j \) are mutually controllable if

1. \( L(G_i)(\Sigma_{uc} \cap \Sigma_j) \cap p_j^i((p_j^k)^{-1}(L(G_k))) \subseteq L(G_i) \)
2. \( L(G_i)(\Sigma_{uc} \cap \Sigma_k) \cap p_k^i((p_k^j)^{-1}(L(G_j))) \subseteq L(G_k) \)

where \( p^k_j : \Sigma^* \rightarrow \Sigma_i^* \) and \( p^i_j : \Sigma^*_i \rightarrow \Sigma^*_j \).

Mutual controllability ensures that after any execution of the system, the occurrence of a shared uncontrollable event is either allowed by every subsystem which shares it, or it is not allowed by any subsystem.

**Definition 4.2 (Local consistency):** A specification \( \mathcal{K} = \mathcal{K} \) is said to be locally consistent w.r.t. \( \Sigma_{uc} \) and \( L(G_i) \) if for any \( i \) we have: \( \forall s \in K_i^{-1} \) and \( \forall u \in \Sigma_{uc} \) such that \( su \in K_i^{-1} \) and \( uv \in \Sigma_{uc}^* \) it holds that \( sp_i(u)v \in K_i^{-1} \Rightarrow sv \in K_i^{-1} \). Based on the above definitions, it holds that the computation of modular supervisors implementing the supremal partially controllable sublanguages of \( K_i^{-1} \) is equivalent to the monolithic supervisor for the specification \( \mathcal{K} \).

**Theorem 4.1 (Supervisor Computation [5]):** Let \( (G_i)_{1 \leq i \leq n} \) be mutually controllable and assume that \( \forall i,k, \Sigma_{uc} \cap \Sigma_{uc} = \emptyset \). If the specification \( \mathcal{K} = \mathcal{K}_{\Sigma} \subseteq \Sigma^* \) is locally consistent w.r.t. \( \Sigma_{uc} \) and \( L(G_i) \) if \( 1 \leq i \leq n \), then

\[
\bigcap_{i} (K_i^{-1})^{pc} = \kappa_{L(G)}(K \cap L(G), \Sigma_{uc}).
\]

Using the concept of modularity [4], the overall supervisor \( S_i^{-1} : \Sigma^* \rightarrow \Gamma_i \) with \( \Gamma_i := \{ \gamma \subseteq \Sigma|\Sigma_{uc} \subseteq \gamma \} \) and \( L(S_i^{-1}/G_i) = \bigcap (K_i^{-1})^{pc} \) can now be implemented as the intersection of the control actions of the modular supervisors \( S_i^{-1} \) with \( L(S_i^{-1}/G_i^{-1}) = (K_i^{-1})^{pc} \) (see Figure 2).

![Modular architecture](image)

Fig. 2. Modular architecture

However, the supervisors \( S_i^{-1} \) are computed based on the reduced system model. In what follows, we provide an implementation of these supervisors for the original systems. For this purpose, \( p_i(L(S_i^{-1}/G_i^{-1})) \) is used as an approximation of the modular closed-loop behavior \( L(S_i^{-1}/G_i^{-1}) = \Gamma \) projected on the reduced subalgebras \( \Sigma_i \). As shown in the next lemma, \( p_i(L(S_i^{-1}/G_i^{-1})) \) is controllable w.r.t. \( L(G_i) \) and \( \Sigma_{uc} \). Thus it can be enforced by a supervisor for \( G_i \).

**Lemma 4.1:** With the preceding notations, \( p_i(L(S_i^{-1}/G_i^{-1})) \) is controllable w.r.t. \( L(G_i) \) and \( \Sigma_{uc} \).

**Proof:** Let us consider

\[
s \in p_i(L(S_i^{-1}/G_i^{-1})) \subseteq L(G_i) \text{ and } s \in \Sigma_i \text{, it is also true that } p_i(s') \in L(G_i). \text{ This entails that } s' \in L(G_i^{-1}) \text{ and } s' \in L(S_i^{-1}/G_i^{-1}) \subseteq L(G_i^{-1}) \subseteq L(G_i) \text{ and } p_i(s') = s \\
\]

But \( L(S_i^{-1}/G_i^{-1}) \) is controllable w.r.t. \( \Sigma_{uc} \) and \( L(G_i^{-1}) \) since it is partially controllable. Therefore, \( s' \in L(S_i^{-1}/G_i^{-1}) \) and \( p_i(s') = s \in p_i(L(S_i^{-1}/G_i^{-1})) \), which concludes the proof.

Now as \( \forall i, p_i(L(S_i^{-1}/G_i^{-1})) \subseteq L(G_i) \) is controllable w.r.t. \( L(G_i) \) and \( \Sigma_{uc} \), there exist \( n \) supervisors \( (S_i)_{1 \leq i \leq n} \) such that

\[
L(S_i/G_i) = p_i(L(S_i^{-1}/G_i^{-1})).
\]

Based on the results in [12], an admissible supervisor for the original system is given with the consistent implementations \( \tilde{S}_i : \tilde{I}_i \rightarrow \tilde{I}_i \) (see [12]) of the decentralized reduced supervisors \( S_i \). It is defined for \( s \in L(G_i) \) as \( \tilde{S}_i(s) := S_i(p_i(s)) \cup (\tilde{S}_i - S_i) \). Note the equality \( p_i^{uc} L(\tilde{S}_i/G_i) = L(S_i/G_i) \).

Combining the steps described above, the main result of this section can be stated.

**Theorem 4.2:** Recalling that \( \Sigma_s \subseteq \Sigma \) and with the notation from above, the supervisor implementation

\[
L(\tilde{S}/G) = (||L(S_i^{-1}/G_i^{-1})||)_{1 \leq i \leq n}
\]
leads to consistent control of the original system, i.e.
\[
p(L(\bar{S}/\bar{G})) = L(S^{-1}/G),
\]
\[
L(\bar{S}/\bar{G}) \subseteq p^{-1}(K),
\]
where \( p : \Sigma^* \rightarrow \Sigma^* \).

The following Lemma aids the proof of Theorem 4.2.

Lemma 4.2 ([14]): Let \((\lambda_i)_{1 \leq i \leq n}\) be languages over the respective alphabets \(\Sigma_i\). Assume that \(\Sigma_0 \subseteq \cup \Sigma_i\) and \(\cup_{i \neq k} (\Sigma_i \cap \Sigma_k) \subseteq \Sigma_0\) with the natural projections \(p_k : (\cup \Sigma_i)^* \rightarrow \Sigma_0^*\) and \(p'_k : (\Sigma_i \cap \Sigma_0)^* \rightarrow \Sigma_0^*\) for \(i = 1, \ldots, n\).

Then
\[
p_0(||L_i||) = ||p'_i(L_i)||.
\]

Proof of Theorem 4.2:

Now Theorem 4.2 can be proven. Because of Lemma 4.2,
\[
p(L(\bar{S}/\bar{G})) = p((L(S^{-1}/G))(|(L(S_i/\bar{G}_i})) = (L(S^{-1}/G)) \circ p(L(S_i/\bar{G}_i)),
\]
\[
= (L(S^{-1}/G)) \circ \rho_{dec}(L(S_i/\bar{G}_i)).
\]

This can be written as
\[
(||L(S_i^{-1}/G_i^{-1})||)(||L(S_i/\bar{G}_i)||)
\]
with \( ||L(S_i^{-1}/G_i^{-1})|| \). Let \( S_i(\bar{G}_i) = L(S_i/\bar{G}_i) \).

Now according to Lemma 4.1, the previous equation can be rearranged as
\[
p(L(\bar{S}/\bar{G})) = \rho_{dec}(L(S_i/\bar{G}_i)) = ||\rho_{dec}(L(S_i^{-1}/G_i^{-1}))||
\]
Moreover, the supervisor computation implies that
\[
\rho_{dec}(L(S_i^{-1}/G_i^{-1})) \subseteq K. \text{ This means that } L(\bar{S}/\bar{G}) \subseteq p^{-1}(K).
\]

The reduced modular architecture is shown in Figure 3. The control actions of the decentralized supervisors for the original systems evaluate to \( \bar{S}_i(s) = p_i(S^{-1}_i(s)) \cup (\Sigma_i - \Sigma_i) \).

V. STRUCTURAL DECENTRALIZED CONTROL

Consider a concurrent system given by a set of nonblocking decentralized systems \((G_i)_{1 \leq i \leq n}\) (i.e. \( \forall i, L_m(G_i) = L(G_i) \)). It follows that the reduced system models \( G_i \) are also nonblocking. We assume that the specification \( \Sigma \subseteq \Sigma^* \) over the subalphabet \(\Sigma\) is separable, i.e. \( K = ||p_i(K)\), where \( p_i : \Sigma^* \rightarrow \Sigma_i^* \) and each local specification \( K_i := p_i(K) \) is \( L_m(G_i)\)-closed, i.e. \( K_i \in F_{L_m(G_i)} \). Our aim is to use the methodology of [8] to compute nonblocking decentralized supervisors acting upon the subsystems \( G_i \) and to implement these supervisors for the original system \( G = ||G_i|| \).

We first formally describe the approach in [8] and then provide new results that are useful in our setting.

Definition 5.1: Let \( \Sigma \subseteq \Sigma \) and \( H \subseteq \Sigma^* \), then \( H \) marks \( \Sigma \) whenever \( \Sigma \Sigma' \cap \Sigma = \emptyset \).

Using the above definition combined with mutual controllability, Theorem 5.1 follows. The structural decentralized architecture is illustrated in Figure 4.

Theorem 5.1 ([8]): Let \((G_i)_{1 \leq i \leq n}\) be nonblocking subsystems and \( K = ||K_i|| \) be the separable specification where \( K_i \in F_{L_m(G_i)} \). Suppose that for \( i, k \leq n \) and \( i \neq k \), \( L_m(G_i) \) and \( L_m(G_k) \) marks \( \Sigma_i \cap \Sigma_k \) and \( L_m(G_i) \) marks the same set, \(^4\) and \( G_i \) and \( G_k \) are mutually controllable, then

1. \( ||K_i \cup K_k|| \cap L(G) = K_i L(G) \).
2. \( p_i(K_i) \) is nonblocking with respect to \( L_m(G_i) \).

In addition to this result due to [8], one can prove that the overall closed-loop behavior is actually nonblocking. To do so, we first show that whenever the local specifications \( K_i \) are \( L_m(G_i)\)-closed then so is the global specification \( K \) with respect to the reduced plant \( G \).

Lemma 5.1: Let \((G_i)_{1 \leq i \leq n}\) be the set of decentralized subsystems and \( K_i \in F_{L_m(G_i)} \). Then \( K = ||K_i|| \in F_{L_m(G)} \).

Proof: First we clearly have that \( \Sigma \subseteq K \). Now since \( \forall i, K_i \subseteq L_m(G_i) \), it holds that \( ||K_i|| \subseteq ||L_m(G_i)|| \) which entails that \( K \subseteq L_m(G) \). Therefore \( K \subseteq K \cap L_m(G) \).

Reciprocally, consider \( s \in K \cap L_m(G) \). We thus have that \( p_i(s) \in p_i(K) \) and \( p_i(s) \in L_m(G) \).

- As \( L_m(G) = \cap_j (p_j^{-1}(L_m(G_j))) \), this entails that \( p_i(s) \in p_i(p_i^{-1}(L_m(G_i))) = L_m(G_i) \).
- Let us now show that \( p_i(s) \in K_i \). First, we have that \( K_i = \cap_j (p_j^{-1}(K_j)) \subseteq \cap_j (p_i^{-1}(K_j)) \subseteq p_i^{-1}(K_i) \).

Also \( p_i(s) \in p_i(K) \subseteq p_i(\cap_j (p_j^{-1}(K_j))) \subseteq \cap_j (p_i(p_j^{-1}(K_j))) \), and hence \( p_i(s) \in p_i(p_i^{-1}(K_i)) = K_i \).

\(^4\)Note that it is equivalent to say that \( \forall i, L_m(G_i) \) marks \( \Sigma_k \).

Fig. 3. Reduced modular architecture

Fig. 4. Structural decentralized architecture
Overall, \( \forall i, \ p_i(s) \in K_i \cap L_m(G_i) = K_i \) as \( K_i \in \mathcal{F}_m(G) \). Thus \( \forall i, \ s \in p_i^{-1}(K_i) \) and \( s \in \cap_i(p_i^{-1}(K_i)) = K. \)

We now need to show that the behavior of the closed-loop reduced system can be actually obtained by a collection of supervisors each of them acting upon a local decentralized subsystem \( G_i \). This is the aim of the next lemma:

**Lemma 5.2:** With the preceding notations, we have that

1) \( || \kappa_{L(G)}(K_i) || = \kappa_{L(G)}(K) \)

2) \( || \kappa_{L(G)}(K_i) || = \kappa_{L(G)}(K) \)

**Proof:**

1) Due to Theorem 5.1, we have that

\[
\kappa_{L(G)}(K) = \| \kappa_{L(G)}(K_i) \cap L(G) \|
\]

Due to the reduced alphabets in addition to the conditions which
mentation, and investigate two different sets of conditions with respect to the reduced system models for the original
sufficient conditions under which a concurrent system is nonblocking.

**Theorem 5.3:** Let \( K \) be a separable specification and let \( S_i \) be such that \( L_m(S_i/G_i) = \kappa_{L(G)}(p_i(K), \Sigma_{dec}) \). Assume that \( G_i \) and \( G_k \) are mutually controllable for \( i \neq k \) and \( L_m(G_i) \) marks \( \Sigma_i \) for all \( i = 1, \ldots, n \). If the supervisors \( S_i \) are consistent implementations of \( S_i \), then the overall supervisor \( \hat{S} \) such that

\[
L(S/G) := ||\|L(S_i/G_i)\|
\]


is nonblocking and consistent.

First we need the following lemma.

**Lemma 5.3** ([12]): The consistent implementation implies that if \( s_j \in L(S_i/G_i) \) and \( s_m u_i \in L(G_i) \) for \( u_i \in (\Sigma_i - \Sigma_i^*) \), then \( s_m u_i \in L(S_i/G_i) \). If additionally \( s_m u_i \in L(G_i) \) for \( u_i \in S_i \) and \( p_i^{dec}(s_m u_i) \in L(S_i/G_i) \), then \( s_m u_i \in L(S_i/G_i) \).

Based on this lemma, the proof of Theorem 5.3 is as follows:

**Proof:** For showing consistency, we observe that

\[
L(S/G) = \|L(S_i/G_i)\| = \|p_i^{dec}(S_i/G_i)\| = \|L(S_i/G_i)\| = L(S/G).
\]

Also, because of \( L(S/G) \subseteq K \), it holds that \( L(S_i/G_i) \subseteq p^{-1}(K) \).

For proving nonblocking control, it has to be shown that

if \( s \in L(S_i/G_i) \), then \( s \in L_m(S_i/G_i) \). Now assume that

\( s \in L(S_i/G_i) \). Then \( \bar{p}_i(s) \in L(S_i/G_i) \) for all \( i = 1, \ldots, n \).

Suppose that there is no \( u_i \in (\Sigma_i - \Sigma_i^*) \) s.t. \( \bar{p}_i(u_i) = L(G_i) \). As \( G_i \) is nonblocking, there must be a string \( \bar{v}_i = \bar{v}_i v_i \in (\Sigma_i - \Sigma_i^*) \Sigma_i^* \) s.t. \( \bar{p}_i(v_i) = L_m(G_i) \). But as \( L_m(G_i) \) marks \( \Sigma_i \), \( \bar{p}_i(\bar{v}_i) = L_m(G_i) \), which contradicts the assumption. As \( \bar{v}_i \) was chosen arbitrarily, it is true that \( \forall i \), there is a \( u_i \in (\Sigma_i - \Sigma_i^*) \) s.t. \( \bar{p}_i(u_i) \in L_m(G_i) \). Hence, for example the string \( s_1 u_1 \cdots u_n \in \|\|\bar{p}_i(u_i)\|\| \subseteq L_m(G_i) = \bar{L}(G_i) \). Now using Lemma 5.3, we also have that \( \bar{p}_i(u_i) \in L(S_i/G_i) \), \( \forall i \), which entails that \( s_1 u_1 \cdots u_n \in \|\|\bar{p}_i(u_i)\|\| \subseteq L_m(G_i) = \bar{L}(S_i/G_i) \).

Thus \( s_1 u_1 \cdots u_n \in L_m(G) \cap L(S/G) = L_m(S/G) \) and thus \( s \in L_m(S/G) \).

The second case is based on the notion of an \( H \)-observer.

**Definition 5.2** (\( H \)-observer): Let \( H \subseteq L = \Sigma^* \subseteq \Sigma^* \) be languages and \( p : \Sigma^* \rightarrow \Sigma^* \) be the natural projection on the alphabet \( \Sigma \subseteq \Sigma^* \). \( p \) is called an \( H \)-observer if \( \forall s \in L \land \forall \sigma \in (\Sigma \cup \{\varepsilon\}) : \)

\[
p(s) \sigma \in p(H) \Rightarrow \exists u \in \Sigma^* \text{ s.t. } u \in H \land p(u) = p(s) \sigma.
\]

In Theorem 5.4, the condition that all events in \( \Sigma_i \) must mark \( L_m(G_i) \) is reduced to the events in \( \Sigma_{dec} \). This is compensated by requiring the decentralized projection \( p_i^{dec} \) to be a \( L_m(G_i) \)-observer.

**Theorem 5.4:** Let \( K \) be a separable specification and let \( S_i \) be such that \( L_m(S_i/G_i) = \kappa_{L(G)}(p_i(K), \Sigma_{dec}) \). Assume that \( G_i \) and \( G_k \) are mutually controllable for \( i \neq k \) and \( L_m(G_i) \) marks \( \Sigma_i \). If \( p_i^{dec} \) is a \( L_m(G_i) \)-observer and the supervisors \( S_i \) are consistent implementations of \( S_i \), then the overall supervisor \( \hat{S} \) such that \( L(S/G) := ||\|L(S_i/G_i)\|| \) is nonblocking and consistent.

---

5If \( p_i^{dec} \) is not a \( L_m(G_i) \)-observer, then [16] provides an algorithm to compute a \( L_m(G_i) \)-observer with the coarsest equivalence kernel possible.
Lemma 5.4 supports the proof of Theorem 5.4.

Lemma 5.4, with the assumptions in Theorem 5.4, it holds that if \( s \in L(\bar{S}_i/\bar{G}_i) \) and \( p_i^{\text{dec}}(s) t \in L_m(\bar{S}_i/\bar{G}_i) \) for \( t \in \Sigma_i^* \), then \( \exists u_t \in \bar{\Sigma}_i^* \) s.t. \( su_t \in L_m(\bar{S}_i/\bar{G}_i) \) and \( p_i^{\text{dec}}(su_t) = p_i^{\text{dec}}(s) t \).

**Proof:** Assume that \( s \in L(\bar{S}_i/\bar{G}_i) \) and \( p_i^{\text{dec}}(s) t \in L_m(\bar{S}_i/\bar{G}_i) \) for \( t \in \Sigma_i^* \). There are two cases.

1. \( t = e \). As \( p_i^{\text{dec}} \) is a \( L_m(\bar{G}_i) \)-observer, there is a \( u_t \in (\bar{\Sigma}_i - \Sigma_i)^* \) s.t. \( su_t \in L_m(\bar{G}_i) \). Because of Lemma 5.3, \( su_t \in L(\bar{S}_i/\bar{G}_i) \) s.t. \( s \in L(\bar{S}_i/\bar{G}_i) \cap L_m(\bar{G}_i) = L_m(\bar{S}_i/\bar{G}_i) \). Consistency follows from the proof of Theorem 5.3. Lemma 5.4 supports the proof of Theorem 5.4.

2. \( t = \sigma_1 \cdots \sigma_m \). As \( p_i^{\text{dec}} \) is a \( L_m(\bar{G}_i) \)-observer, there is a \( u_t = v_0 \sigma_1 v_1 \cdots \sigma_{m-1} v_{m-1} \cdot \sigma_m^0 \in \bar{\Sigma}_i^* \) s.t. \( su_t \in L_m(\bar{G}_i) \) and \( p_i^{\text{dec}}(u_t) = t \), i.e., \( v_j \in (\bar{\Sigma}_i - \Sigma_i)^* \) for all \( j = 0, \ldots, m \). Successive application of Lemma 5.3 implies \( su_t \in L(\bar{S}_i/\bar{G}_i) \). Thus, \( su_t \in L(\bar{S}_i/\bar{G}_i) \cap L_m(\bar{G}_i) = L_m(\bar{S}_i/\bar{G}_i) \).

**Proof of Theorem 5.4:**

Consistency follows from the proof of Theorem 5.3. Now assume that \( s \in L(\bar{S}_i/\bar{G}_i) \). Then \( s_i := \bar{p}_i(s) \in L(\bar{S}_i/\bar{G}_i) \) and \( p_i^{\text{dec}}(s) \in L(\bar{S}_i/\bar{G}_i) \). As \( S_i \) is a nonblocking supervisor, there is a string \( t \in \Sigma_i^* \) s.t. \( p_i^{\text{dec}}(s) t \in L_m(\bar{S}_i/\bar{G}_i) \) and s.t. all its predecessors are not marked, i.e., \( \forall t' < t \) we have that \( p_i^{\text{dec}}(s) t' \not\in L(\bar{S}_i/\bar{G}_i) \). Then it holds that \( t \in (\bar{\Sigma}_i - \Sigma_i)^* \) (otherwise there would be a marked predecessor string as \( L_m(\bar{G}_i) \) marks \( \Sigma_i \)). Because of Lemma 5.4, there is a \( u_t \in \Sigma_i^* \) s.t. \( su_t \in L_m(\bar{S}_i/\bar{G}_i) \) and \( p_i^{\text{dec}}(su_t) = p_i^{\text{dec}}(s) t \). Furthermore, as \( p_i^{\text{dec}}(u_t) = t \subseteq (\bar{\Sigma}_i - \Sigma_i)^s \), it turns out that \( u_t \in (p_i^{\text{dec}})^{-1}(t) \subseteq (\bar{\Sigma}_i - \Sigma_i)^s \). As \( i \) was arbitrary, such \( u_t \) exists for all \( i = 1, \ldots, n \). Hence, for example the string \( su_1 \cdots u_n \in \Sigma_i \) s.t. \( su_t \subseteq L_m(\bar{S}_i/\bar{G}_i) \) and consequently \( s \in L_m(\bar{S}_i/\bar{G}_i) \).

The reduced structural decentralized control architecture is depicted in Figure 5.

![Diagram of the reduced decentralized control architecture](image-url)

**VI. CONCLUSIONS**

We have developed two methods exploiting the structure of concurrent systems for the supervisor synthesis without composition of the overall plant. In our approach, the computational complexity is further reduced by using reduced system models for supervisor computation. Our modular approach can be applied to prefix-closed non-separable specifications and results in modular supervisors in a conjunctive architecture. Additionally, we elaborated a decentralized approach which is feasible for specifications that are separable but not necessarily prefix-closed. We provide two different sets of conditions which guarantee nonblocking control of the original system. It has to be noted that although maximally permissive supervisors could be computed for the reduced system models, the supervisors for the original system need not be maximally permissive. In further work, we want to investigate conditions which also guarantee maximally permissive supervisors for the original system.

**REFERENCES**