On Maximal Permissiveness of Hierarchical and Modular Supervisory Control Approaches for Discrete Event Systems

Klaus Schmidt, Christian Breindl

Abstract—Recently, several efficient modular and hierarchical approaches for the control of discrete event systems (DES) have been proposed. Although these methods are very suitable for dealing with the state space explosion problem, their common limitation is that either maximal permissiveness is not addressed or unnecessarily restrictive conditions are required in order to ensure maximally permissive control. In this paper we develop a unified framework for the investigation of maximal permissiveness of modular and hierarchical supervisory control. We first introduce local control consistency (OCC) as a condition for maximal permissive control in monolithic two-level hierarchies as in [1], [2], [3], and show that OCC is less conservative than LCC. Then, we extend our framework to modular control and multi-level hierarchies, and prove that the additional requirement of mutual controllability (see [8]) is sufficient for maximal permissiveness of the methods in [3], [4], [5], [6], [7].

The remainder of the paper is organized as follows. Section II provides basic definitions of the supervisory control theory. In Section III, we investigate maximal permissiveness for monolithic hierarchical control, and in Section IV we elaborate our main result on maximal permissiveness for modular control in multi-level hierarchies. Conclusions are given in Section V.

I. INTRODUCTION

In recent years, a variety of approaches that reduce the computational effort of the supervisor synthesis for discrete event systems (DES) by employing modular and hierarchical control techniques has been developed. As a common feature, the methods in [1], [2], [3], [4], [5], [6], [7] use the natural projection that exhibits the observer property in order to determine abstracted system models that enable efficient computations on smaller state spaces. Furthermore, except for [1], they are designed for systems that are composed of modular components.

The important result of the above approaches is that the supervisors determined for both the modular system components and the abstracted system models can be implemented in a modular fashion, hence avoiding the enumeration of the overall system state space. Moreover, it is guaranteed that the modular supervisors are nonblocking and comply with the specified system behavior, i.e., if such modular supervisors have been found, it is guaranteed that the closed-loop system fulfills the specification and does not block. On the downside, maximal permissiveness is not ensured by most of the above approaches. There might exist a less restrictive monolithic nonblocking supervisor meeting the given specification. In particular, it can happen that no modular supervisors can be found although there exists a modular solution. A first result towards maximally permissive modular and hierarchical control has been obtained in [4], where output control consistency (OCC) is required. As will be shown in Section III-C, OCC can be replaced by a less restrictive condition.

In this paper, we propose a unified framework for the investigation of maximal permissiveness in modular and hierarchical supervisory control. We first introduce local control consistency (LCC) as a condition for maximal permissive control in monolithic two-level hierarchies as in [1], [2], [3], and show that LCC is less conservative than OCC. Then, we extend our framework to modular control and multi-level hierarchies, and prove that the additional requirement of mutual controllability (see [8]) is sufficient for maximal permissiveness of the methods in [3], [4], [5], [6], [7].
set of all control patterns is denoted \( \Gamma \subseteq 2^\Sigma \). A supervisor is a map \( S \colon L(G) \to \Gamma \), where \( S(s) \) represents the set of enabled events after the occurrence of string \( s \). The language \( L(S(G)) \) generated by \( G \) under supervision is \( S \) is iteratively defined by (1) \( e \in L(S(G)) \) and (2) \( s\sigma \in L(S(G)) \) iff \( s \in L(S(G)), \sigma \in S(s) \) and \( \sigma \in L(G) \). Thus, \( L(S(G)) \) represents the behavior of the closed-loop system.

A language \( M \) is said to be controllable w.r.t. \( L(G) \) if \( \overline{M} \Sigma_{\text{uc}} \cap L(G) \subseteq \overline{M} \). The set of all languages that are controllable w.r.t. \( L(G) \) is denoted \( \mathcal{C}(L(G)) \). Furthermore, the set \( \mathcal{C}(L(G)) \) is closed under union. In particular, for every specification language \( E \) there uniquely exists a supremal controllable sublanguage of \( E \) w.r.t. \( L(G) \), which is formally defined as \( \kappa_{L(G)}(E) := \bigcup \{ M \in \mathcal{C}(L(G)) \mid M \subseteq E \} \). To take into account \( E \) and the marking of \( G \) in the closed-loop behavior, we employ a marking supervisor as in [10] s.t. \( L_m(S/G) := L(S/G) \). A supervisor \( S \) that leads to a closed-loop behavior \( \kappa_{L(G)}(E \cap L_m(S/G)) \) is said to be maximally permissive.

A widely used property in the context of hierarchical supervisory control is the observer property.

**Definition 1 (I):** Let \( M' \subseteq M \subseteq \Sigma^* \) be languages and let \( P_0 : \Sigma^* \to \Sigma \) be the natural projection for \( \Sigma^\ast \subseteq \Sigma^* \). \( P_0 \) is an \( M' \)-observer (w.r.t. \( M \)) iff for all \( s \in \overline{M} \) and \( t \in \Sigma^\ast \), \( P_0(s) \in P_0(M') \Rightarrow \exists \exists u \in \Sigma^* \text{ s.t. } su \in M' \wedge P_0(su) = P_0(s)t. 

**III. MONOLITHIC HIERARCHICAL ARCHITECTURE**

**A. Control Architecture**

In this section, we elaborate how maximally permissive control can be achieved in the architecture in Fig. 1. Definition 2 gives a detailed description of this architecture.

**Definition 2 (Monolithic Architecture):** The following entities and conditions are required.

(i) **low level:** the plant is modeled by an automaton \( G \) with the alphabet \( \Sigma \). The uncontrollable events are \( \Sigma_{\text{uc}} \subseteq \Sigma \), and there is a low-level supervisory control \( S : \Sigma^* \to \Gamma \).

(ii) **high level:** the high-level alphabet is \( \Sigma^h \subseteq \Sigma \) with the uncontrollable high-level events \( \Sigma_{\text{uc}}^h := \Sigma^h \cap \Sigma_{\text{uc}} \), and the uncontrollable language \( L^h = \langle \Sigma^h \rangle \). The high-level plant \( G^h \) is determined by \( L(G^h) \) and \( L_m^h := \langle \Sigma^h \rangle \). There is a high-level supervisor \( \Gamma^h : \langle \Sigma^h \rangle \to \Gamma^h \) defined by \( \Gamma^h := \{ \gamma | \gamma \Sigma_{\text{uc}}^h \subseteq \gamma \subseteq \Sigma^h \} \).

(iii) **supervisor computation:** for the specification \( K^h \subseteq \langle \Sigma^h \rangle \), the high-level supervisor is computed such that \( L_m^h \langle \Sigma^h \rangle \subseteq \kappa_{L(G)}(K^h) \subseteq \langle \Sigma^h \rangle \). The low-level supervisor fulfills \( L_m(S(G)) \subseteq L_m^h \langle \Sigma^h \rangle \subseteq \langle \Sigma^h \rangle \), i.e., \( S \) is efficiently implemented based on \( \Sigma^h, \Sigma, \Sigma^h \).

(iv) **nonblocking control:** we require the low-level control is nonblocking, i.e., \( L_m(S(G)) = L(S(G)) \).

(v) **abstraction condition:** the natural projection \( p^h \) is an \( L(G) \)-observer for \( L(G) \) according to Definition 1.

In this section, we assume that a hierarchical architecture with the above features is given. Note that several approaches such as [1] and [2] indeed comply with Definition 2, where

\[ (v) \text{ is common to all approaches, and different conditions provide nonblocking control in } (iv). \]

\[ \text{Fig. 1. Monolithic hierarchical control architecture} \]

Having described the control architecture, we now discuss sufficient conditions for maximally permissive hierarchical control. With \( K := L_m(S/G) \) as the specification for the low-level plant, it is desired that \( L_m(S/G) = \kappa_{L(G)}(K) =: L_max \). We first state a result that is the basis of our investigation. Lemma 1 reduces the verification of maximal permissiveness to a controllability test.

**Lemma 1:** Assume the control architecture in Definition 2 (i)-(iv). Then, it holds that \( S \) is maximally permissive if \( p^h(L_{\text{max}}) \) is controllable w.r.t. \( L(G^h) \), i.e.,

\[ p^h(L_{\text{max}}) \subseteq L(G^h) \subseteq p^h(L_{\text{max}}) \Rightarrow L_m(S/G) = L_{\text{max}} \]

**Proof:** As \( L(S/G) \) is controllable w.r.t. \( L(G) \), the fact that \( S \) is nonblocking establishes that \( L_m(S/G) \) is controllable w.r.t. \( L(G) \). Together with \( L_m(S/G) \subseteq L_m(G) \), this implies that \( L_m(S/G) \subseteq L_{\text{max}} \).

To show the reverse inclusion, we observe that

\[ p^h(L_{\text{max}}) \subseteq L(G^h) \subseteq p^h(L_{\text{max}}) \]

and \( L_{\text{max}} \subseteq p^h((p^h(L_{\text{max}}))^{-1} \subseteq L_m(S/G) \), then

\[ L_{\text{max}} \subseteq p^h(L_{\text{max}}) \cap L_{\text{max}} \]

is controllable w.r.t. \( L(G^h) \). Since \( L_{\text{max}} \subseteq \langle p^h \rangle^{-1}(p^h(L_{\text{max}})) \) and \( L_{\text{max}} \subseteq L_m(G) \), also

\[ L_{\text{max}} \subseteq \langle p^h \rangle^{-1}(p^h(L_{\text{max}})) \cap L_m(G) = p^h(L_{\text{max}}) \cap L_m(G) \subseteq L_m(S/G) \cap L_m(S/G) \]

is controllable w.r.t. \( L(G^h) \).

**B. Output Control Consistency**

In hierarchical supervisory control, output control consistency (OCC) is used as a condition to ensure maximal permissiveness. It has been first stated for general causal reporter maps in [11], and then formulated for natural projections in [4].

**Definition 3 (OCC [4]):** Let \( M = \overline{M} \subseteq \Sigma^* \) be a prefix-closed language, and let \( \Sigma_{\text{uc}} \subseteq \Sigma \) and \( \Sigma^h \subseteq \Sigma \) be the set of uncontrollable and high-level events, respectively. The natural projection \( p^h : \Sigma^* \to \langle \Sigma^h \rangle \) is output control consistent (occ) for \( M \) if for every \( s \in M \) of the form

\[ s = \sigma_1 \cdots \sigma_k \text{ or } s = s' \sigma_0 \sigma_1 \cdots \sigma_k, \quad k \geq 1, \]

where \( \sigma_0, \sigma_k \in \Sigma^h \) and \( \sigma_i \in \Sigma - \Sigma^h \) for \( i = 1, \ldots, k-1 \), we have the property that \( \sigma_i \in \Sigma_{\text{uc}} \Rightarrow (\forall i = 1, \ldots, k) \sigma_i \in \Sigma_{\text{uc}} \).

This means that, whenever \( \sigma_i \) is an uncontrollable high-level event, its immediately preceding low-level events must all be uncontrollable, such that its nearest controllable event is a high-level event.

In this section, we show that adding OCC to the architecture in Definition 2 results in maximally permissive control.

**Theorem 1:** Assume the control architecture in Definition 2. If \( p^h \) is occ for \( L(G) \), then \( S \) is maximally permissive.
implies that \( s_{hi} \) denote the architecture in Definition 2 (such as [1], [2]) is extended and pursued in [13], [14]. Theorem 4 (LCC): Let \( G \) be an automaton, and \( \Sigma_{hi} \) its hierarchical abstraction with the corresponding high-level alphabet \( \Sigma^h \) and natural projection \( p^h : \Sigma^* \rightarrow (\Sigma^h)^* \). We denote \( p^h \) locally control consistent (lcc) for a string \( s \in L(G) \) if for all \( \sigma_{uc} \in \Sigma_{uc} \) s.t. \( p^h(\sigma_{uc}) \in L(G^h) \), it holds that either \( \exists u \in (\Sigma - \Sigma^h)^* \) s.t. \( \sigma_{uc} \) is controllable w.r.t. \( L(G) \) or there is a controllable \( u \in (\Sigma_{uc} - \Sigma^h)^* \) s.t. \( \sigma_{uc} \in L(G) \). Furthermore, we call \( p^h \) lcc for all \( s \in M \). In words, a natural projection is locally control consistent if for each uncontrollable high-level event \( \sigma_{uc} \) that is feasible after the corresponding high-level string, there is either no continuation or an uncontrollable continuation of \( s \) that terminates with \( \sigma_{uc} \). Hence, if \( \sigma_{uc} \) is possible after \( s \), then it cannot be an uncontrollable continuation.

Based on Definition 4, we can replace OCC in Theorem 1 by LCC for a certain set of strings.

Theorem 2 (LCC): Assume the control architecture in Definition 2. Let \( s_{hi} \in L(G^h) \), and define \( L_{en}(s_{hi}) := \{ s \in L(G) \mid p^h(s) = s_{hi} \wedge \exists s' < s \text{ s.t. } p^h(s') = s_{hi} \} \) as the set of shortest possible low-level strings that are projected to \( s_{hi} \).

To show that OCC implies LCC, let \( p^h \) be occ, and assume that \( s \in L_{en}(s_{hi}) \) for some \( s_{hi} \in L(G^h) \) s.t. \( s_{hi} \in L(G^h) \) for some \( \sigma_{uc} \in \Sigma_{uc} \). It holds that either \( \exists u \in (\Sigma - \Sigma^h)^* \) s.t. \( \sigma_{uc} \) is controllable w.r.t. \( L(G) \) or there is a \( u \in (\Sigma_{uc} - \Sigma^h)^* \) s.t. \( \sigma_{uc} \in L(G) \). The above result demonstrates that LCC indeed provides a less conservative condition than OCC in order to ensure maximal permissiveness for control. For a comparison of LCC and OCC, consider the automaton \( G \) in Fig. 2 with the high-level alphabet \( \Sigma^h = \{ \alpha, \beta \} \) and the controllable events \( \Sigma_{ec} = \{ a, d, e, h \} \) (represented by ticks on arrows in the figure). Observing that all strings in \( L(G^h) \) with the uncontrollable successor event \( \beta \) reach the state 2, the only state that is reached by corresponding entry strings in \( L(G) \) is 2. As \( u = ab \in (\Sigma_{uc} - \Sigma^h)^* \) and \( \delta(2, ab\beta) \) exists, it follows that the natural projection \( p^h : \Sigma^* \rightarrow (\Sigma^h)^* \) is lcc for \( L_{en}(s_{hi}) \) for all \( s_{hi} \in L(G^h) \). Note that \( p^h \) is not occ as for example \( u = cde \not\in (\Sigma_{uc} - \Sigma^h)^* \) but \( \delta(2, u') \) exists.

Additionally, it has to be noted that the construction of a natural projection \( p^h \) that is an \( L(G) \) observer and fulfills LCC for \( L_{en}(s_{hi}) \) can be formulated as a coarsest relational partition problem (CRPP) as in [12], [13]. This is a relevant result, since recent studies show that computing natural projections \( p^h \) that are sufficient for nonblocking control in [1], [2] can also be stated as CRPP [13], [14]. Hence, it is possible to determine the natural projection \( p^h \) with the coarsest equivalence kernel that guarantees nonblocking control and maximal permissiveness in polynomial time.

IV. HIERARCHICAL AND MODULAR CONTROL

A. Control Architecture

In this section, we consider the hierarchical and modular architecture depicted in Fig. 3. The major difference with respect to the architecture in Section III-A is that now the plant is no longer represented by a single automaton but composed of a set of automata.

Definition 5 (Modular Architecture): The following entities and conditions are required.

(i) low level: the plant is modeled by \( n \) automata \( G_i \), \( i = 1, \ldots, n \) with the respective alphabet \( \Sigma_i \). The overall plant is \( G := \bigcup_{i=1}^{n} G_i \) over \( \Sigma := \bigcup_{i=1}^{n} \Sigma_i \). The uncontrollable events are given as \( \Sigma_{i, uc} \subseteq \Sigma_i \) such that \( \Sigma_{uc} := \bigcup_{i=1}^{n} \Sigma_{i, uc} \), and there is a low-level supervisor \( S : \Sigma^* \rightarrow \Gamma \).
B. Maximal Permissiveness for Modular Control

In the modular case, a slightly more restrictive condition that is still less conservative than OCC is required to ensure maximally permissive control, i.e., with \( K := L_m(G) \cap \Sigma_i^{\text{hi}} \), \( L_m(S/G) = \kappa_{L(G)}(K) \).

Theorem 3 (LCC Modular): Assume the control architecture described in Definition 5. For \( i = 1, \ldots, n \), let \( s_i^{\text{hi}} \in L(G_i^{\text{hi}}) \), and define \( L_{\text{loc},i}(s_i^{\text{hi}}) := \{ s \in L(G_i) \mid p_{\text{dec}}(s) = s_i^{\text{hi}} \} \) as the set of strings in \( L(G_i) \) that are projected to \( s_i^{\text{hi}} \). If for all \( i = 1, \ldots, n \), and for all \( s_i^{\text{hi}} \in L(G_i^{\text{hi}}) \), \( p_{\text{dec}} \) is lcc for \( L_{\text{loc},i}(s_i^{\text{hi}}) \), then \( S \) is maximally permissive.

**Proof:** It can be verified that all conditions in Definition 2 hold. In particular, with [4], \( p_i^{\text{hi}} : \Sigma^* \to (\Sigma_i^{\text{hi}})^* \) is an \( L(G_i) \)-observer. Hence, it is sufficient to show that the conditions in Theorem 3 imply that \( p_i^{\text{hi}} \) is lcc for \( L_{\text{en}}(s_i^{\text{hi}}) \) for all \( s_i^{\text{hi}} \in L(G_i^{\text{hi}}) \), i.e., the conditions in Theorem 2 are fulfilled.

Let \( s_i^{\text{hi}} \in L(G_i^{\text{hi}}) \) and \( s_i^{\text{hi}} \in \Sigma_i^{\text{hi}} \) s.t. \( s_i^{\text{hi}} \sigma_i \in L(G_i^{\text{hi}}) \). Then, there exists a \( s \in L_{\text{en}}(s_i^{\text{hi}}) \). For \( i = 1, \ldots, n \), we define the natural projections \( p_i : \Sigma^* \to \Sigma_i^* \) and \( p_i^{\text{hi}} : (\Sigma_i^{\text{hi}})^* \to (\Sigma_i^{\text{hi}})^* \). With \( s_i^{\text{hi}} := p_i^{\text{hi}}(s_i^{\text{hi}}) \), it holds for all \( i = 1, \ldots, n \) that \( s_i := p_i(s) \in L_{\text{loc},i}(s_i^{\text{hi}}) \). Furthermore, for all \( i \) s.t. \( \sigma_i \in \Sigma_i^{\text{hi}} \), \( s_i^{\text{hi}} \sigma_i \in L(G_i^{\text{hi}}) \). As \( p_{\text{dec}} \) is an \( L(G_i) \)-observer and lcc for \( L_{\text{loc},i}(s_i^{\text{hi}}) \), for all \( i \) such that \( \sigma_i \in \Sigma_i \), there exists a \( u_i \in (\Sigma_i^{\text{hi}} \cap \Sigma_i^{\text{hi}})^* \) s.t. \( s_i u_i \sigma_i \in L(G_i) \). For all remaining \( i \), let \( u_i = \epsilon \). Defining \( u := u_1 \cdots u_n, u \in (\sum_i^{\text{hi}})^\ast \), implies that \( s u \sigma_i \in L(G) = (\sum_i^{\text{hi}})^\ast \). Thus \( \sigma_i \sigma_i \in (\Sigma_i^{\text{hi}})^\ast \). Since \( p_i^{\text{hi}} = L_{\text{en}}(s_i^{\text{hi}}) \) are arbitrary, this proves that \( p_i^{\text{hi}} \) is lcc for \( L_{\text{en}}(s_i^{\text{hi}}) \) for all \( s_i^{\text{hi}} \in L(G_i^{\text{hi}}) \).

At this point, it has to be noted that the construction of a natural projection \( p_{\text{dec}}^{\text{hi}} \) that is an \( L(G_i) \)-observer and fulfills LCC for \( L_{\text{loc},i}(s_i^{\text{hi}}) \) can also be formulated as a CRPP. Hence, similar to Section III-C, \( p_{\text{dec}}^{\text{hi}} \) can be computed such that it supports nonblocking controller synthesis and maximal permissiveness for the approaches in [2], [6], [7].

C. Maximal Permissiveness with Local Control

In this section, we choose the same setup as in Fig. 3 with the modification that now each modular component \( G_i \) represents a local control system \( S_i/H_i \). In the following definition, we state the extensions to Definition 5.

**Definition 6 (Modular Architecture):**

(i) **low level:** the plant is represented by \( n \) automata \( H_i \), \( i = 1, \ldots, n \) with the respective alphabet \( \Sigma_i \), and the overall plant is \( H := \bigcap_{i=1}^n H_i \) over \( \Sigma := \bigcup_{i=1}^n \Sigma_i \). There are local supervisors \( S_i : \Sigma_i^* \to \Gamma_i := (\gamma_{i} / \Sigma_i^{\text{uc}} \subseteq \gamma \in \Sigma_i) \). Furthermore, it holds that \( G_i = S_i / H_i \).

(ii) **high level:** identical to Definition 5.

(iii) **supervisor computation:** each local supervisor is determined as \( L_m(S_i / H_i) = \kappa_{L(H_i)}(L_m(H_i) / \Gamma_i) \). Then, the low-level supervisor is implemented as \( L_m(S/G) = L_m(S/H) \).

(iv) **nonblocking control:** we require that the low-level control is nonblocking, i.e., \( L_m(S/G) = L(S/G) \).

(v) **abstraction condition:** we require that the natural projection \( p_{\text{dec}}^{\text{hi}} \) is an \( L(G) \)-observer for \( L(G_i) \), \( i = 1, \ldots, n \).

Again, it is sufficient to ensure nonblocking control by assumption. Therefore, it is possible to achieve a unified treatment of maximal permissiveness for all approaches that comply with these requirements such as [2], [6], [7].

---

**Fig. 3. Hierarchical and modular control architecture**
Extending the conditions in Theorem 3 with mutual controllability is sufficient for maximally permissible control considering that local control is applied.

**Theorem 4 (LCC Modular):** Assume the control architecture in Definition 6. If for all \( i = 1, \ldots, n \) and for all \( s_{hi} \in L(G_{hi}), \mu^b_{hi} \) is lcc for \( L_{unc}(s_{hi}) \), and all local components \( H_i, H_k, i = 1, \ldots, n, i \neq k \), are mutually controllable, then \( S \) is maximally permissible.

The proof of Theorem 4 relies on the following lemmas.

**Lemma 2 (Lemma A.8 in [3]):** Let \( H := \biguplus_{i=1}^n H_i \), let \( H_i, H_k, i = 1, \ldots, n, i \neq k \), be mutually controllable, and let \( s_i \in L(H_i) \) and \( s \in \Sigma_{unc}^i \) s.t. \( s_i \in L(H_i) \). Then, for all \( s \in L(H) \) s.t. \( p_i(s) = s_i \), it holds that \( s \in L(H) \).

**Lemma 3:** Let \( H_i, H_k, i = 1, \ldots, n, i \neq k \), be mutually controllable. Then, \( \kappa_{L(H)}(L_m(H)) = \bigcup_{i=1}^n \kappa_{L(H)}(L_m(H_i)) \).

**Proof:** Let \( s \in L(H) \). Then, for some \( k, s_k \in \kappa_{L(H)}(L_m(H_k)) \), i.e., there is a \( u_k \in \Sigma_{unc}^i \) s.t. \( s_k u_k \notin \Sigma_{k} \) but \( s_k u_k \in L(H_k) \). Let \( u_k = v_i \gamma_{i} \) where \( v_i \in \Sigma_{unc}^i \), \( j = 1, \ldots, m + 1 \) and \( j \neq m \). Since of mutual controllability, repeated application of Lemma 2 yields \( s_k u_k \in L(H) \) but \( s_k u_k \notin \bigcup_{i=1}^n \kappa_{L(H)} \) since \( s_k u_k \notin \Sigma_{k} \). This violates the assumption \( s \in \kappa_{L(H)}(L_m(H)) \), and hence, \( s \notin L_m(H) \).

**Lemma 4 (Exercise 3.7.13 in [10]):** Let \( H \) be an automaton over \( \Sigma \), and \( K_1, K_2 \subseteq \Sigma \) be specifications. Then, \( \kappa_{L(H)}(L_m(H)) = \kappa_{L(H)}(L_m(H)) \).

With Lemma 3 and Lemma 4, Theorem 4 can be proved.

**Proof:** \( \kappa_{L(H)}(L_m(H)) \) can be computed from \( L_m(H) \) by evaluating \( \kappa_{L(G)}(L_m(H)) \), i.e., \( \kappa_{L(G)}(L_m(H)) \) is \( \kappa_{L(H)}(L_m(H)) \). To see this, first assume that \( s \in \kappa_{L(G)}(L_m(H)) \). Then, because of Lemma 3, we have that \( s \in L_m(H) \). From this and the fact that \( L(G) \subseteq L(H) \) it follows that \( s \in \kappa_{L(G)}(L_m(H)) \). To show the other inclusion let \( s \in \kappa_{L(G)}(L_m(H)) \). Since \( L_m(H) \subseteq \bigcup_{i=1}^n \kappa_{L(H)}(L_m(H_i)) \) and every entry that is controllable w.r.t. \( L(G) \) is also controllable w.r.t. \( L(H) \), we obtain \( s \in \kappa_{L(G)}(L_m(H)) \).

Now first Lemma 4, and then Theorem 3 are applied to obtain \( \kappa_{L(G)}(L_m(H)) \).

**D. Modular Multi-Level Hierarchy**

In this section we show that the conditions in Theorem 4 are also sufficient for multi-level hierarchical architectures. To this end, we present a generic three-level hierarchy as in Fig. 4, and elaborate an inductive argument that can be carried over to multi-level hierarchies.

**Definition 8 (Multi-Level Architecture):** The following entities and conditions are required.

- **First level:** The plant is represented by \( n \) groups of automata \( H_{1,i} \), \( i = 1, \ldots, n \), \( k = 1, \ldots, n_i \) with the respective alphabet \( \Sigma_{1,i}^{k}\). Hence, the plant for each group is \( H_{1,i} := \bigcup_{k=1}^{n_i} H_{1,i}^{k} \) over \( \Sigma_{1,i}^{k} \) and the overall plant is \( H := \bigcup_{i=1}^{n} \bigcup_{k=1}^{n_i} H_{1,i}^{k} \).

- **Second level:** The set of shared events \( \Sigma_{1,i}^{k} \) of all plants on the first level, the second-level alphabet fulfills \( \Sigma_{1,i}^{k} \subseteq \Sigma_{2,i} \subseteq \Sigma \). There are \( n \) components \( H_{2,i} = \bigcup_{k=1}^{n_i} H_{2,i}^{k} \) over \( \Sigma_{2,i} \) where \( \Sigma_{2,i} \subseteq \Sigma_{1,i}^{k} \). The uncontrollable events are given as \( \Sigma_{unc}^{i} \subseteq \Sigma_{2,i} \) such that \( \Sigma_{unc}^{i} \subseteq \bigcup_{i=1}^{n} \bigcup_{k=1}^{n_i} \Sigma_{unc}^{i,k} \). There are local supervisors \( S_{1,i}^{k} : (\Sigma_{1,i}^{k})^{*} \rightarrow \Gamma_{1,i}^{k} \) and low-level supervisors \( S_{2,i}^{k} : (\Sigma_{2,i})^{*} \rightarrow \Gamma_{2,i}^{k} \).

- **Third level:** The set of shared events \( \Sigma_{2,i} \) of all plants on the second level, it holds that \( \Sigma_{2,i} \subseteq \Sigma_{3,i} \subseteq \Sigma \). For each second-level component, we have \( \Sigma_{3,i} := \Sigma_{2,i} \cap \Sigma_{3,i} \) and \( \Sigma_{unc}^{i,c} := \Sigma_{2,i} \cap \Sigma_{unc}^{i} \). Furthermore, the natural projection \( p_{2,i}^{k} : (\Sigma_{2,i})^{*} \rightarrow (\Sigma_{3,i})^{*} \) is defined.

- **Supervisor computation:** each first-level supervisor is determined as \( L_{m}(S_{1,i}^{k} H_{1,i}^{k}) = \kappa_{L_{m}(H_{1,i}^{k})}(L_{m}(H_{1,i}^{k})) \) for specifications \( H_{1,i}^{k} \subseteq \bigcup_{i=1}^{n} \bigcup_{k=1}^{n_i} H_{1,i}^{k} \), \( i = 1, \ldots, n, k = 1, \ldots, n_i \). The second-level supervisors are computed as...
nonblocking control in hierarchical and modular control architectures. It can be result to a hierarchy with an arbitrary number of levels as system components do not share events.

In this regard, note that a projection that complies applied to the approaches in [2], [4], [5], [7] as long as the abstraction in the 3-level hierarchy according to Definition 8 is maximally permissive.

Theorem 5 (Multi-Level Architecture): Assume the control architecture in Definition 8. We require that for all $i=1,\ldots,n$, $k=1,\ldots,n_i$ it holds that for all $s^i\in L(G^i)$, $p^i_k$ is lcc for $L^{loc,i}(s^i)$, and for all $s^i\in L(G^i)$, $p^i_k$ is lcc for $L^{loc,i,k}(s^i)$. Furthermore let $H^i_{1,k}$ and $H^i_{2,l}$ be mutually controllable for all $i,j=1,\ldots,n$ and $k,l=1,\ldots,n_i$, $l=1,\ldots,n_j$ such that $i\neq j$ or $k\neq l$, and $H^i_{2,l}$ be mutually controllable for all $i,j=1,\ldots,n$, $i\neq j$. Then $S$ is maximally permissive.

Proof: Let $K^2 := L_m(H^2)/\|K^2\|_2/\|K^2\|_1/\|K^2\|_{n,n_2}$ and $K^1 := L_m(H^1)/\|K^1\|_2/\|K^1\|_1/\|K^1\|_{n,n_1}$. Appealing to Theorem 4, we know that (i) $L_m(S^2/G^2) = \kappa_{L(H^2)}(K^2) = L_m(G^2)$ (situation in Fig. 4 (a)) and (ii) $\kappa_{L(H^1)}(K^1) = \kappa_{L(H^2)}(K^2)/L_m(G^1)$ (situation in Fig. 4 (b)). Combining (i) and (ii), we arrive at $\kappa_{L(H^1)}(K^1) = L_m(G^3)/L_m(G^2)/L_m(G^1)$.

With this argument, it is straightforward to transfer the result to a hierarchy with an arbitrary number of levels as long as the conditions in Definition 8 are met for each three consecutive levels. To our knowledge, Theorem 5 is the most general result concerning maximal permissiveness in hierarchical and modular control architectures. It can be applied to the approaches in [2], [4], [5], [7] as long as the additional requirements of LCC and mutual controllability are met. In this regard, note that a projection that complies with LCC can be computed algorithmically, and mutual controllability trivially holds for the methods in [4], [5] as system components do not share events.

V. Conclusions

In this paper, a unified framework for studying maximal permissiveness in modular and hierarchical supervisory control has been proposed. It has been designed to incorporate hierarchical control approaches that employ natural projections with the observer property for system abstraction such as [1], [2], [3], [4], [5], [6], [7]. In this framework, we first introduced local control consistency as a less conservative condition for maximally permissive monolithic hierarchical control. This result was then extended to modular and multi-level hierarchical control, where mutual controllability was found as an additional condition to ensure maximal permissiveness for the methods in [2], [3], [4], [5], [6], [7].

In our investigations, it turned out that the computation of a natural projection that exhibits local control consistency can be formulated as a coarsest relational partition problem similar to the computation of natural projections that support nonblocking control in the above approaches. Algorithms for the computation of natural projections that are both suitable for nonblocking control and fulfill local control consistency have been developed and first applied in [15].

References