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 control in multi-level hierarchies. We identify a set of conditions that is met by several approaches, and prove its sufficiency for maximally permissive control.**

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 monolithic 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].

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.

II. Preliminaries

We recall basics from supervisory control theory [9], [10].

For a finite alphabet Σ, the set of all finite strings over Σ is denoted Σ^*. We write $s_1 s_2 \in \Sigma^*$ for the concatenation of two strings $s_1,\ s_2 \in \Sigma^*$, and $s_1 \le s$ when s_1 is a *prefix* of s, i.e. if there exists a string $s_2 \in \Sigma^*$ with $s = s_1 s_2$. The empty string is denoted $\epsilon \in \Sigma^*$, i.e. $s\epsilon = \epsilon s = s$ for all $s \in \Sigma^*$. A *language* over Σ is a subset $M \subseteq \Sigma^*$. The *prefix closure* of M is defined by $\overline{M} := \{s_1 \in \Sigma^* | \, \exists s \in M \text{ s.t. } s_1 \le s\}$. A language M is *prefix closed* if $M = \overline{M}$.

The *natural projection* $p_i : \Sigma^* \to \Sigma_i^*$, $i = 1, 2$, for the (not necessarily disjoint) union $\Sigma = \Sigma_1 \cup \Sigma_2$ is defined iteratively: (1) let $p_i(\epsilon) := \epsilon$; (2) for $s \in \Sigma^*$, $\sigma \in \Sigma$, let $p_i(s\sigma) := p_i(s)\sigma$ if $\sigma \in \Sigma_i$, or $p_i(s\sigma) := p_i(s)$ otherwise. The set-valued inverse of p_i is denoted $p_i^{-1} : \Sigma_i^* \to 2^{\Sigma^*}$. The *synchronous product* $M_1 || M_2 \subseteq \Sigma^*$ of two languages $M_i \subseteq \Sigma_i^*$ is $M_1 || M_2 = p_1^{-1}(M_1) \cap p_2^{-1}(M_2) \subseteq \Sigma^*$.

A *finite automaton* is a tuple $G = (X, \Sigma, \delta, x_0, X_\mathrm{m})$, with the finite set of *states* X; the finite alphabet of *events* Σ; the partial *transition function* $\delta : X \times \Sigma \to X$; the *initial state* $x_0 \in X$; and the set of *marked states* $X_\mathrm{m} \subseteq X$. We write $\delta(x, \sigma)!$ if δ is defined at (x, σ). In order to extend δ 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 $\delta(x', \sigma)!$. $L(G) := \{s \in \Sigma^* : \delta(x_0, s)!\}$ and $L_\mathrm{m}(G) := \{s \in L(G) : \delta(x_0, s) \in X_\mathrm{m}\}$ are the *closed* and *marked language* generated by the finite automaton G, respectively. A formal definition of the synchronous composition $G_1 || G_2$ of two automata G_1 and G_2 can be taken from e.g. [9]. Note that $L_\mathrm{m}(G_1 || G_2) = L_\mathrm{m}(G_1) || L_\mathrm{m}(G_2)$.

In a supervisory control context, we write $\Sigma = \Sigma_\mathrm{c} \dot\cup \Sigma_\mathrm{uc}$ to distinguish *controllable* (Σ_c) and *uncontrollable* (Σ_uc) events. A *control pattern* is a set γ, $\Sigma_\mathrm{uc} \subseteq \gamma \subseteq \Sigma$, and the

K. Schmidt is with the Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg, Germany, klaus.schmidt@rt.eei.uni-erlangen.de

C. Breindl is with the Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany breindl@ist.uni-stuttgart.de

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 S is iteratively defined by (1) $\epsilon \in L(S/G)$ and (2) $s\sigma \in L(S/G)$ iff $s \in L(S/G), \sigma \in S(s)$ and $s\sigma \in L(G)$. 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 $\overline{M}\Sigma_{\rm uc} \cap L(G) \subseteq \overline{M}$. 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 arbitrary 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 $\kappa_{L(G)}(E) := \cup\{M \in \mathcal{C}(L(G))|\ M \subseteq E\}$. 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. $L_{\rm m}(S/G) := \kappa_{L(G)}(E \cap L_{\rm m}(G))$. Then, the closed-loop system is *nonblocking*, i.e., $\overline{L_{\rm m}(S/G)} = L(S/G)$. A supervisor S that leads to a closed-loop behavior $\kappa_{L(G)}(E \cap L_{\rm m}(G))$ is said to be *maximally permissive*.

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

Definition 1 ([1]): Let $M' \subseteq M \subseteq \Sigma^*$ be languages and let $p_0 : \Sigma^* \to \Sigma_0^*$ be the natural projection for $\Sigma_0 \subseteq \Sigma$. p_0 is an M'-observer (w.r.t. M) iff for all $s \in \overline{M}$ and $t \in \Sigma_0^*$

$$p_0(s)t \in p_0(M') \Rightarrow \exists u \in \Sigma^* \text{ s.t. } su \in M' \wedge p_0(su) = p_0(s)t.$$

III. Monolithic Hierarchical Control

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 Σ. The uncontrollable events are $\Sigma_{\rm uc} \subseteq \Sigma$, and there is a low-level supervisor $S : \Sigma^* \to \Gamma$.

(ii) *high level*: the high-level alphabet is $\Sigma^{\rm hi} \subseteq \Sigma$ with the uncontrollable high-level events $\Sigma_{\rm uc}^{\rm hi} := \Sigma^{\rm hi} \cap \Sigma_{\rm uc}$ and the natural projection $p^{\rm hi} : \Sigma^* \to (\Sigma^{\rm hi})^*$. The high-level plant $G^{\rm hi}$ is determined by $L(G^{\rm hi}) := p^{\rm hi}(L(G))$ and $L_{\rm m}(G^{\rm hi}) := p^{\rm hi}(L_{\rm m}(G))$, and there is a high-level supervisor $S^{\rm hi} : (\Sigma^{\rm hi})^* \to \Gamma^{\rm hi} := \{\gamma|\Sigma_{\rm uc}^{\rm hi} \subseteq \gamma \subseteq \Sigma^{\rm hi}\}$.

(iii) *supervisor computation*: for the specification $K^{\rm hi} \subseteq (\Sigma^{\rm hi})^*$, the high-level supervisor is computed such that $L_{\rm m}(S^{\rm hi}/G^{\rm hi}) = \kappa_{L(G^{\rm hi})}(L_{\rm m}(G^{\rm hi})||K^{\rm hi})$. The low-level supervisor fulfills $L_{\rm m}(S/G) = L_{\rm m}(S^{\rm hi}/G^{\rm hi})||L_{\rm m}(G)$, i.e., S is efficiently implemented based on $S^{\rm hi}$, Σ, $\Sigma^{\rm hi}$.

(iv) *nonblocking control*: we require that the low-level control is nonblocking, i.e., $\overline{L_{\rm m}(S/G)} = L(S/G)$.

(v) *abstraction condition*: the natural projection $p^{\rm hi}$ 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][1] and [2] indeed comply with Definition 2, where (v) is common to all approaches, and different conditions provide nonblocking control in (iv).

[1]Here, the natural projection has to be used as the causal reporter map.

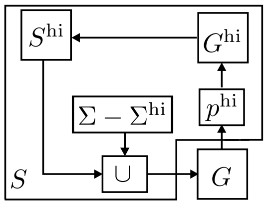

Fig. 1. Monolithic hierarchical control architecture

Having described the control architecture, we now discuss sufficient conditions for maximally permissive hierarchical control. With $K := L_{\rm m}(G)||K^{\rm hi}$ as the specification for the low-level plant, it is desired that $L_{\rm m}(S/G) = \kappa_{L(G)}(K) =: L_{\max}$. 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 $p^{\rm hi}(L_{\max})$ is controllable w.r.t. $L(G^{\rm hi})$, i.e.,

$$\overline{p^{\rm hi}(L_{\max})}\Sigma_{\rm uc}^{\rm hi} \cap L(G^{\rm hi}) \subseteq \overline{p^{\rm hi}(L_{\max})} \Rightarrow L_{\rm m}(S/G) = L_{\max}$$

Proof: As $L(S/G)$ is controllable w.r.t. $L(G)$, the fact that S is nonblocking establishes that $L_{\rm m}(S/G)$ is controllable w.r.t. $L(G)$. Together with $L_{\rm m}(S/G) \subseteq L_{\rm m}(G)||K^{\rm hi}$, this implies that $L_{\rm m}(S/G) \subseteq L_{\max}$.

To show the reverse inclusion, we observe that $p^{\rm hi}(L_{\max})||L_{\rm m}(G) \subseteq L_{\rm m}(S^{\rm hi}/G^{\rm hi})||L_{\rm m}(G)$ as $p^{\rm hi}(L_{\max})$ is controllable w.r.t. $L(G^{\rm hi})$. Since $L_{\max} \subseteq (p^{\rm hi})^{-1}(p^{\rm hi}(L_{\max}))$ and $L_{\max} \subseteq L_{\rm m}(G)$, also $L_{\max} \subseteq (p^{\rm hi})^{-1}(p^{\rm hi}(L_{\max})) \cap L_{\rm m}(G) = p^{\rm hi}(L_{\max})||L_{\rm m}(G) \subseteq L_{\rm m}(S^{\rm hi}/G^{\rm hi})||L_{\rm m}(G)$. ■

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 $\Sigma_{\rm uc} \subseteq \Sigma$ and $\Sigma^{\rm hi} \subseteq \Sigma$ be the set of uncontrollable and high-level events, respectively. The natural projection $p^{\rm hi} : \Sigma^* \to (\Sigma^{\rm hi})^*$ is output control consistent (occ) for M if for every $s \in M$ of the form

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

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

This means that, whenever σ_k 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 $p^{\rm hi}$ is occ for $L(G)$, then S is maximally permissive.

Proof: Appealing to Lemma 1, it has to be shown that $p^{\mathrm{hi}}(L_{\max})$ is controllable w.r.t. $L(G^{\mathrm{hi}})$.

Assume that $s^{\mathrm{hi}} \in L(G^{\mathrm{hi}}) \cap \overline{p^{\mathrm{hi}}(L_{\max})}$, and let $\sigma_{\mathrm{uc}} \in \Sigma_{\mathrm{uc}}^{\mathrm{hi}}$ s.t. $s^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in L(G^{\mathrm{hi}})$. It has to be shown that $s^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in \overline{p^{\mathrm{hi}}(L_{\max})}$. As $s^{\mathrm{hi}} \in \overline{p^{\mathrm{hi}}(L_{\max})} = p^{\mathrm{hi}}(\overline{L_{\max}})$, there is a $s \in \overline{L_{\max}}$ s.t. $p^{\mathrm{hi}}(s) = s^{\mathrm{hi}}$. Then, $s \in L(G)$, and because of Definition 2 (v), there exists a $u \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L(G)$. Writing $s = s'\sigma u'$ with $\sigma \in \Sigma^{\mathrm{hi}}$, $u' \in (\Sigma - \Sigma^{\mathrm{hi}})^*$, and considering Definition 3, it is readily observed that $u'u \in (\Sigma_{\mathrm{uc}} - \Sigma_{\mathrm{uc}}^{\mathrm{hi}})^*$. Hence, $su\sigma_{\mathrm{uc}} \in \overline{L_{\max}}$ and consequently $s^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in p^{\mathrm{hi}}(\overline{L_{\max}}) = \overline{p^{\mathrm{hi}}(L_{\max})}$. ■

Theorem 1 states that if any approach that complies with the architecture in Definition 2 (such as [1], [2]) is extended with the requirement for OCC, then maximally permissive control is achieved. This is of particular interest, as there are polynomial time algorithms that refine a natural projection p^{hi} in order to fulfill OCC [11], [10].

C. Local Control Consistency

OCC requires that any local string between a high-level event and an uncontrollable high-level event has to contain only uncontrollable low-level events. In this section, we relax this assumption to *local control consistency* (LCC), while still guaranteeing maximal permissiveness in the framework of Definition 2.

Definition 4 (LCC): Let G be a finite automaton, and G^{hi} its hierarchical abstraction with the corresponding high-level alphabet Σ^{hi} and natural projection $p^{\mathrm{hi}} : \Sigma^* \to (\Sigma^{\mathrm{hi}})^*$. We denote p^{hi} *locally control consistent* (lcc) for a string $s \in L(G)$ if for all $\sigma_{\mathrm{uc}} \in \Sigma_{\mathrm{uc}}^{\mathrm{hi}}$ s.t. $p^{\mathrm{hi}}(s)\sigma_{\mathrm{uc}} \in L(G^{\mathrm{hi}})$, it holds that either $\nexists u \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L(G)$ or there is a $u \in (\Sigma_{\mathrm{uc}} - \Sigma_{\mathrm{uc}}^{\mathrm{hi}})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L(G)$. Furthermore, we call p^{hi} lcc for a language $M \subseteq L(G)$ if p^{hi} is lcc for all $s \in M$.

In words, a natural projection is locally control consistent for a string $s \in L(G)$, if for each uncontrollable high-level event σ_{uc} that is feasible after the corresponding high-level string, there is either no continuation or an uncontrollable continuation of s that terminates with σ_{uc}. Hence, if σ_{uc} is possible after s, then it cannot be prevented.

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 $s^{\mathrm{hi}} \in L(G^{\mathrm{hi}})$, and define $L_{\mathrm{en}}(s^{\mathrm{hi}}) := \{s \in L(G) | p^{\mathrm{hi}}(s) = s^{\mathrm{hi}} \wedge \nexists s' < s \text{ s.t. } p^{\mathrm{hi}}(s') = s^{\mathrm{hi}}\}$ as the set of shortest possible low-level strings that are projected to s^{hi}. If for all $s^{\mathrm{hi}} \in L(G^{\mathrm{hi}})$, p^{hi} is lcc for $L_{\mathrm{en}}(s^{\mathrm{hi}})$, then S is maximally permissive. Furthermore, OCC implies LCC.

Proof: Again, it has to be shown that $p^{\mathrm{hi}}(L_{\max})$ is controllable w.r.t. $L(G^{\mathrm{hi}})$. Let $s^{\mathrm{hi}} \in p^{\mathrm{hi}}(\overline{L_{\max}})$, and assume that $\sigma_{\mathrm{uc}} \in \Sigma_{\mathrm{uc}}$ s.t. $s^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in L(G^{\mathrm{hi}})$. It has to be proved that $s^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in p^{\mathrm{hi}}(\overline{L_{\max}})$. As $s^{\mathrm{hi}} \in p^{\mathrm{hi}}(\overline{L_{\max}})$, there must be an $s \in L_{\mathrm{en}}(s^{\mathrm{hi}}) \cap \overline{L_{\max}}$. Now it suffices to show that there is a continuation $u \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma_{\mathrm{uc}} \in \overline{L_{\max}}$. As p^{hi} is an $L(G)$-observer and lcc for $L_{\mathrm{en}}(s^{\mathrm{hi}})$, there is an uncontrollable continuation $u \in (\Sigma_{\mathrm{uc}} - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L(G)$. Then, the fact that $L_{\max}$ is controllable w.r.t. $L(G)$ implies that $su\sigma_{\mathrm{uc}} \in \overline{L_{\max}}$.

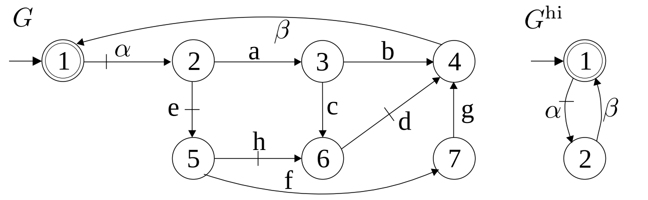

Fig. 2. Comparison of OCC and LCC

To show that OCC implies LCC, let p^{hi} be occ, and assume that $s \in L_{\mathrm{en}}(s^{\mathrm{hi}})$ for some $s^{\mathrm{hi}} \in L(G^{\mathrm{hi}})$ s.t. $s^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in L(G^{\mathrm{hi}})$ for some $\sigma_{\mathrm{uc}} \in \Sigma_{\mathrm{uc}}$. It holds that either $\nexists u \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L(G)$ or there is a $u \in (\Sigma - \Sigma^{\mathrm{hi}})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L(G)$. In the first case, LCC holds by definition. Otherwise, since p^{hi} is occ, it must hold that $u \in (\Sigma_{\mathrm{uc}} - \Sigma_{\mathrm{uc}}^{\mathrm{hi}})^*$. Noting that s^{hi}, σ_{uc}, and $s \in L_{\mathrm{en}}(s^{\mathrm{hi}})$ were chosen arbitrarily, p^{hi} is lcc. ■

The above result demonstrates that LCC indeed provides a less conservative condition than OCC in order to ensure maximally permissive control. For a comparison of LCC and OCC, consider the automaton G in Fig. 2 with the high-level alphabet $\Sigma^{\mathrm{hi}} = \{\alpha, \beta\}$ and the controllable events $\Sigma_{\mathrm{c}} = \{\alpha, d, e, h\}$ (represented by ticks on arrows in the figure). Observing that all strings in $L(G^{\mathrm{hi}})$ with the uncontrollable successor event β reach the state 2, the only state that is reached by corresponding entry strings in $L(G)$ is 2. As $u = ab \in (\Sigma_{\mathrm{uc}} - \Sigma_{\mathrm{uc}}^{\mathrm{hi}})^*$ and $\delta(2, ab\beta)$ exists, it follows that the natural projection $p^{\mathrm{hi}} : \Sigma^* \to (\Sigma^{\mathrm{hi}})^*$ is lcc for $L_{\mathrm{en}}(s^{\mathrm{hi}})$ for all $s^{\mathrm{hi}} \in L(G^{\mathrm{hi}})$. Note that p^{hi} is not occ as for example $u' = ehd \notin (\Sigma_{\mathrm{uc}} - \Sigma_{\mathrm{uc}}^{\mathrm{hi}})^*$ but $\delta(2, u'\beta)$ exists.

Additionally, it has to be noted that the construction of a natural projection p^{hi} that is an $L(G)$-observer and fulfills LCC for $L_{\mathrm{en}}(s^{\mathrm{hi}})$ 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^{hi} 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^{hi} 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 Σ_i. The overall plant is $G := ||_{i=1}^n G_i$ over $\Sigma := \bigcup_{i=1}^n \Sigma_i$. The uncontrollable events are given as $\Sigma_{i,\mathrm{uc}} \subseteq \Sigma_i$ such that $\Sigma_{\mathrm{uc}} := \bigcup_{i=1}^n \Sigma_{i,\mathrm{uc}}$, and there is a low-level supervisor $S : \Sigma^* \to \Gamma$.

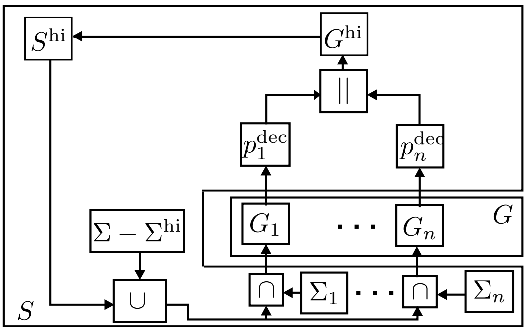

Fig. 3. Hierarchical and modular control architecture

(ii) *high level*: With the set of *shared events* $\Sigma_\cap := \bigcup_{i,j=1,i\neq j}^{n}(\Sigma_i \cap \Sigma_j)$, the high-level alphabet fulfills $\Sigma_\cap \subseteq \Sigma^{\mathrm{hi}} \subseteq \Sigma$. For each component, we have $\Sigma_i^{\mathrm{hi}} := \Sigma_i \cap \Sigma^{\mathrm{hi}}$ and $\Sigma_{i,\mathrm{uc}}^{\mathrm{hi}} := \Sigma_i^{\mathrm{hi}} \cap \Sigma_{i,\mathrm{uc}}$, $i = 1,\ldots,n$. Furthermore, the natural projection $p_i^{\mathrm{dec}} : \Sigma_i^* \to (\Sigma_i^{\mathrm{hi}})^*$ is defined. The high-level plant G_i^{hi} for each component is determined by $L(G_i^{\mathrm{hi}}) := p_i^{\mathrm{dec}}(L(G_i))$ and $L_{\mathrm{m}}(G_i^{\mathrm{hi}}) := p_i^{\mathrm{dec}}(L_{\mathrm{m}}(G_i))$. The overall high-level plant evaluates to $G^{\mathrm{hi}} := ||_{i=1}^{n} G_i^{\mathrm{hi}}$ with the uncontrollable high-level events $\Sigma_{\mathrm{uc}}^{\mathrm{hi}} := \Sigma^{\mathrm{hi}} \cap \Sigma_{\mathrm{uc}}$, and there is a high-level supervisor $S^{\mathrm{hi}} : (\Sigma^{\mathrm{hi}})^* \to \Gamma^{\mathrm{hi}} := \{\gamma | \Sigma_{\mathrm{uc}}^{\mathrm{hi}} \subseteq \gamma \subseteq \Sigma^{\mathrm{hi}}\}$.

(iii) *supervisor computation*: for the specification $K^{\mathrm{hi}} \subseteq (\Sigma^{\mathrm{hi}})^*$, first the high-level supervisor is computed such that $L_{\mathrm{m}}(S^{\mathrm{hi}}/G^{\mathrm{hi}}) = \kappa_{L(G^{\mathrm{hi}})}(L_{\mathrm{m}}(G^{\mathrm{hi}})||K^{\mathrm{hi}})$. Then, the low-level supervisor is implemented as $L_{\mathrm{m}}(S/G) = L_{\mathrm{m}}(S^{\mathrm{hi}}/G^{\mathrm{hi}})||L_{\mathrm{m}}(G)$.

(iv) *nonblocking control*: we require that the low-level control is nonblocking, i.e., $\overline{L_{\mathrm{m}}(S/G)} = L(S/G)$.

(v) *abstraction condition*: we require that the natural projection p_i^{dec} is an $L(G_i)$-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].

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 $K := L_{\mathrm{m}}(G) \cap K^{\mathrm{hi}}$, $L_{\mathrm{m}}(S/G) = \kappa_{L(G)}(K)$.

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

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

Let $s^{\mathrm{hi}} \in L(G^{\mathrm{hi}})$ and $\sigma_{\mathrm{uc}} \in \Sigma_{\mathrm{uc}}$ s.t. $s^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in L(G^{\mathrm{hi}})$. Then, there exists a $s \in L_{\mathrm{en}}(s^{\mathrm{hi}})$. For $i = 1,\ldots,n$, we define the natural projections $p_i : \Sigma^* \to \Sigma_i^*$ and $p_i^{\mathrm{hi}} : (\Sigma^{\mathrm{hi}})^* \to (\Sigma_i^{\mathrm{hi}})^*$. With $s_i^{\mathrm{hi}} := p_i^{\mathrm{hi}}(s^{\mathrm{hi}})$, it holds for all $i = 1,\ldots,n$ that $s_i := p_i(s) \in L_{\mathrm{loc},i}(s_i^{\mathrm{hi}})$. Furthermore, for all i s.t. $\sigma_{\mathrm{uc}} \in \Sigma_i$, $s_i^{\mathrm{hi}}\sigma_{\mathrm{uc}} \in L(G_i^{\mathrm{hi}})$. As p_i^{dec} is an $L(G_i)$-observer and lcc for $L_{\mathrm{loc,i}}(s^{\mathrm{hi}})$, for all i such that $\sigma_{\mathrm{uc}} \in \Sigma_i$, there exists a $u_i \in (\Sigma_{i,\mathrm{uc}} - \Sigma_{i,\mathrm{uc}}^{\mathrm{hi}})^*$ s.t. $s_i u_i \sigma_{\mathrm{uc}} \in L(G_i)$. For all remaining i, let $u_i = \epsilon$. Defining $u := u_1 \ldots u_n$, $u \in ||_{i=1}^{n}\{u_i\}$ implies that $su\sigma_{\mathrm{uc}} \in L(G) = ||_{i=1}^{n} L(G_i)$, and $u \in (\Sigma_{\mathrm{uc}} - \Sigma_{\mathrm{uc}}^{\mathrm{hi}})^*$. Since s^{hi}, σ_{uc} and $s \in L_{\mathrm{en}}(s^{\mathrm{hi}})$ were arbitrary, this proves that p^{hi} is lcc for $L_{\mathrm{en}}(s^{\mathrm{hi}})$ for all $s^{\mathrm{hi}} \in L(G^{\mathrm{hi}})$. ■

At this point, it has to be noted that the construction of a natural projection p_i^{dec} that is an $L(G_i)$-observer and fulfills LCC for $L_{\mathrm{loc,i}}(s_i^{\mathrm{hi}})$ can also be formulated as a CRPP. Hence, similar to Section III-C, p_i^{dec} can be computed such that it both 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 Σ_i, and the overall plant is $H := ||_{i=1}^{n} H_i$ over $\Sigma := \bigcup_{i=1}^{n}\Sigma_i$. There are local supervisors $S_i : \Sigma_i^* \to \Gamma_i := \{\gamma | \Sigma_{i,\mathrm{uc}} \subseteq \gamma \subseteq \Sigma_i\}$. 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 $L_{\mathrm{m}}(S_i/H_i) = \kappa_{L(H_i)}(L_{\mathrm{m}}(H_i)||K_i)$. Then, the low-level supervisor is implemented as $L_{\mathrm{m}}(S/H) = L_{\mathrm{m}}(S^{\mathrm{hi}}/G^{\mathrm{hi}})||L_{\mathrm{m}}(S_1/H_1)|| \cdots ||L_{\mathrm{m}}(S_n/H_n)$.

(iv) *nonblocking control*: $\overline{L_{\mathrm{m}}(S/H)} = L(S/H)$.

(v) *abstraction condition*: identical to Definition 5.

Definition 6 describes a situation of high practical interest. In addition to the high-level specification K^{hi} that usually states the desired cooperative behavior of several system components, local specifications K_i for individual components are given. Using local control by the supervisors S_i avoids the composition of the overall low-level plant. In Fig. 3, the described change amounts to replacing G_i by S_i/H_i. Note that in the case with local specifications, we have $K := L_{\mathrm{m}}(H)||K^{\mathrm{hi}}||K_1|| \cdots ||K_n$, and the maximally permissive control is given by $L_{\max} := \kappa_{L(H)}(K)$.

Different from the situation in Theorem 3, we now have to account for the possibility that the local control by the supervisors S_i, $i = 1,\ldots,n$ is more restrictive than a maximally permissive supervisor. In this work, we employ *mutual controllability* that was found in [8] as a structural condition that ensures maximal permissiveness of the local control. We first define the alphabets $\Sigma_{i,k} := \Sigma_i \cap \Sigma_k$ for $i,k = 1,\ldots,n$ and $i \neq k$, and the corresponding natural projections $p_{i,k} : \Sigma_i^* \to \Sigma_{i,k}^*$.

Definition 7 (Mutual Controllability [8]): The automata H_i and H_k are mutually controllable if

$$L(H_k)(\Sigma_{i,k} \cap \Sigma_{i,\mathrm{uc}}) \cap (p_{k,i})^{-1}(p_{i,k}(L(H_i))) \subseteq L(H_k),$$
$$L(H_i)(\Sigma_{i,k} \cap \Sigma_{k,\mathrm{uc}}) \cap (p_{i,k})^{-1}(p_{k,i}(L(H_k))) \subseteq L(H_i).$$

Extending the conditions in Theorem 3 with mutual controllability is sufficient for maximally permissive 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 $s^{\mathrm{hi}} \in L(G_i^{\mathrm{hi}})$, p_i^{dec} is lcc for $L_{\mathrm{loc,i}}(s^{\mathrm{hi}})$, and all local components H_i, H_k, $i, k = 1, \ldots, n$, $i \neq k$, are mutually controllable, then S is maximally permissive.

The proof of Theorem 4 relies on the following lemmas.

Lemma 2 (Lemma A.8 in [3]): Let $H := ||_{j=1}^n H_j$, let H_i, H_k, $i, k = 1, \ldots, n$, $i \neq k$, be mutually controllable, and let $s_i \in L(H_i)$ and $\sigma \in \Sigma_{i,\mathrm{uc}}^{\mathrm{hi}}$ s.t. $s_i\sigma \in L(H_i)$. Then, for all $s \in L(H)$ s.t. $p_i(s) = s_i$, it holds that $s\sigma \in L(H)$.

Lemma 3: Let H_i, H_k, $i, k = 1, \ldots, n$, $i \neq k$, be mutually controllable. Then, $\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n) \subseteq L_{\mathrm{m}}(G) = ||_{i=1}^n \kappa_{L(H_i)}(L_{\mathrm{m}}(H_i)||K_i)$.

Proof: Let $s \in \kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n)$. Then, $s \in L_{\mathrm{m}}(H)$, $s_i := p_i(s) \in K_i$ for $i = 1, \ldots, n$, and $\nexists u \in \Sigma_{\mathrm{uc}}^*$ s.t. $su \in L(H)$ and $su \notin \overline{L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n}$.

Now assume that $s \notin L_{\mathrm{m}}(G)$. Then, for some k, $s_k \notin \kappa_{L(H_k)}(L_{\mathrm{m}}(H_k)||K_k)$, i.e., there is a $u_k \in \Sigma_{k,\mathrm{uc}}^*$ s.t. $s_k u_k \notin \overline{K_k}$ but $s_k u_k \in L(H_k)$. Let $u_k = v_1\sigma_1 \cdots v_m\sigma_m v_{m+1}$, where $v_j \in (\Sigma_{k,\mathrm{uc}} - \Sigma_{k,\mathrm{uc}}^{\mathrm{hi}})^*$, $j = 1, \ldots, m+1$ and $\sigma_j \in \Sigma_{k,\mathrm{uc}}^{\mathrm{hi}}$, $j = 1, \ldots, m$. Because of mutual controllability, repeated application of Lemma 2 yields $su_k \in L(H)$ but $su_k \notin \overline{||_{i=1}^n K_i}$ since $s_k u_k \notin \overline{K_k}$. This violates the assumption $s \in \kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n)$, and hence, $s \in L_{\mathrm{m}}(G)$. ■

Lemma 4 (Exercise 3.7.13 in [10]): Let H be an automaton over Σ, and $K_1, K_2 \subseteq \Sigma^*$ be specifications. Then

$$\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1||K_2) = \kappa_{\overline{\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1)}}(\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1)||K_2).$$

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

Proof: $\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n)$ can be computed from $L_{\mathrm{m}}(G)$ by evaluating $\kappa_{L(G)}(L_{\mathrm{m}}(G))$, i.e., $\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n) = \kappa_{L(G)}(L_{\mathrm{m}}(G))$. To see this, first assume that $s \in \kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n)$. Then, because of Lemma 3, we have that $s \in L_{\mathrm{m}}(G)$. From this and the fact that $L(G) \subseteq L(H)$ it follows that $s \in \kappa_{L(G)}(L_{\mathrm{m}}(G))$. To show the other inclusion let $s \in \kappa_{L(G)}(L_{\mathrm{m}}(G))$. Since $L_{\mathrm{m}}(G) \subseteq (L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n)$ and every string that is controllable w.r.t. $L(G)$ is also controllable w.r.t. $L(H)$, we obtain $s \in \kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n)$. With Lemma 4, we can write

$$\kappa_{L(H)}(K) = \kappa_{L(H)}(L_{\mathrm{m}}(H)||K^{\mathrm{hi}}||K_1|| \cdots ||K_n) = \kappa_{\overline{\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1||\cdots||K_n)}}(\kappa_{L(H)}(L_{\mathrm{m}}(H)||K_1|| \cdots ||K_n)|| ||K^{\mathrm{hi}}) = \kappa_{\overline{\kappa_{L(G)}(L_{\mathrm{m}}(G))}}(\kappa_{L(G)}(L_{\mathrm{m}}(G))||K^{\mathrm{hi}}).$$

Now first Lemma 4, and then Theorem 3 are applied to obtain

$$\kappa_{\overline{\kappa_{L(G)}(L_{\mathrm{m}}(G))}}(\kappa_{L(G)}(L_{\mathrm{m}}(G))||K^{\mathrm{hi}}) = \kappa_{L(G)}(L_{\mathrm{m}}(G)||K^{\mathrm{hi}}) = L_{\mathrm{m}}(S^{\mathrm{hi}}/G^{\mathrm{hi}})||L_{\mathrm{m}}(G) = L_{\mathrm{m}}(S^{\mathrm{hi}}/G^{\mathrm{hi}})||L_{\mathrm{m}}(S_1/H_1)|| \cdots ||L_{\mathrm{m}}(S_n/H_n).$$

■

The assumptions in this section are suitable for a two-level hierarchy according to the approach in [3] extended with local control, and for each step in the hierarchical construction in [4], [5]. Note that maximal permissiveness is also investigated in [4]. However, in that paper, OCC is required for the modular components, and it is assumed that the modular components do not share events. Observing that both our LCC condition is less conservative than OCC, and the absence of shared events implies mutual controllability, our result is more general.

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.

(i) *first level*: the plant is represented by n groups of automata $H_{i,k}^1$, $i = 1, \ldots, n$, $k = 1, \ldots, n_i$ with the respective alphabet $\Sigma_{i,k}^1$. Hence, the plant for each group is $H_i^1 := ||_{k=1}^{n_i} H_{i,k}^1$ over $\Sigma_i^1 := \bigcup_{k=1}^{n_i} \Sigma_{i,k}^1$, and the overall plant is $H := ||_{i=1}^n ||_{k=1}^{n_i} H_{i,k}^1$ over $\Sigma := \bigcup_{i=1}^n \bigcup_{k=1}^{n_i} \Sigma_{i,k}^1$. The uncontrollable events are given as $\Sigma_{i,k,\mathrm{uc}}^1 \subseteq \Sigma_{i,k}^1$ such that $\Sigma_{\mathrm{uc}} := \bigcup_{i=1}^n \bigcup_{k=1}^{n_i} \Sigma_{i,k,\mathrm{uc}}^1$. There are local supervisors $S_{i,k}^1 : (\Sigma_{i,k}^1)^* \to \Gamma_{i,k}^1 := \{\gamma | \Sigma_{i,k,\mathrm{uc}}^1 \subseteq \gamma \subseteq \Sigma_{i,k}^1\}$ and low-level supervisors $S_i^1 : (\Sigma_i^1)^* \to \Gamma_i^1 := \{\gamma | \Sigma_{i,\mathrm{uc}}^1 \subseteq \gamma \subseteq \Sigma_i^1\}$. Furthermore, it holds that $G_{i,k}^1 := S_{i,k}^1/H_{i,k}^1$ and $G^1 := ||_{i=1}^n ||_{k=1}^{n_i} G_{i,k}^1$.

(ii) *second level*: With the set of *shared events* $\Sigma_\cap^1$ of all plants on the first level, the second-level alphabet fulfills $\Sigma_\cap^1 \subseteq \Sigma^2 \subseteq \Sigma$. There are n components $H_i^2 = ||_{k=1}^{n_i} H_{i,k}^2$ over $\Sigma_i^2 := \bigcup_{k=1}^{n_i} \Sigma_{i,k}^2$, where $\Sigma_{i,k}^2 := \Sigma_{i,k}^1 \cap \Sigma^2$. The uncontrollable events are $\Sigma_{i,\mathrm{uc}}^2 := \Sigma_i^2 \cap \Sigma_{\mathrm{uc}}$, $i = 1, \ldots, n$. With $p_{i,k}^1 : (\Sigma_{i,k}^1)^* \to (\Sigma_{i,k}^2)^*$, the high-level plant $H_{i,k}^2$ for each component is determined by $L(H_{i,k}^2) := p_{i,k}^1(L(G_{i,k}^1))$ and $L_{\mathrm{m}}(H_{i,k}^2) := p_{i,k}^1(L_{\mathrm{m}}(G_{i,k}^1))$. There is a supervisor $S^2 : (\Sigma^2)^* \to \Gamma^2 := \{\gamma | \Sigma_{\mathrm{uc}}^2 \subseteq \gamma \subseteq \Sigma^2\}$, and for $i = 1, \ldots, n$, there is a supervisor $S_i^2 : (\Sigma_i^2)^* \to \Gamma_i^2 := \{\gamma | \Sigma_{i,\mathrm{uc}}^2 \subseteq \gamma \subseteq \Sigma_i^2\}$. We define $G_i^2 := S_i^2/H_i^2$ and $G^2 := ||_{i=1}^n G_i^2$.

(iii) *third level*: With the set of *shared events* $\Sigma_\cap^2$ of all plants on the second level, it holds that $\Sigma_\cap^2 \subseteq \Sigma^3 \subseteq \Sigma^2$. For each second-level component, we have $\Sigma_i^3 := \Sigma_i^2 \cap \Sigma^3$ and $\Sigma_{i,\mathrm{uc}}^3 := \Sigma_i^3 \cap \Sigma_{i,\mathrm{uc}}^2$, $i = 1, \ldots, n$. Furthermore, the natural projection $p_i^2 : (\Sigma_i^2)^* \to (\Sigma_i^3)^*$ is defined. The high-level plant H_i^3 for each component is determined by $L(H_i^3) := p_i^3(L(G_i^2))$ and $L_{\mathrm{m}}(H_i^3) := p_i^3(L_{\mathrm{m}}(G_i^2))$. The overall high-level plant evaluates to $H^3 := ||_{i=1}^n H_i^3$ with the uncontrollable high-level events $\Sigma_{\mathrm{uc}}^3 := \Sigma^3 \cap \Sigma_{\mathrm{uc}}$, and there is a third-level supervisor $S^3 : (\Sigma^3)^* \to \Gamma^3 := \{\gamma | \Sigma_{\mathrm{uc}}^3 \subseteq \gamma \subseteq \Sigma^3\}$. We define $G^3 := S^3/H^3$.

(iv) *supervisor computation*: each first-level supervisor is determined as $L_{\mathrm{m}}(S_{i,k}^1/H_{i,k}^1) = \kappa_{L(H_{i,k}^1)}(L_{\mathrm{m}}(H_{i,k}^1)||K_{i,k}^1)$ for specifications $K_{i,k}^1 \subseteq (\Sigma_{i,k}^1)^*$, $i = 1, \ldots, n$, $k = 1, \ldots, n_i$. The second-level supervisors are computed as

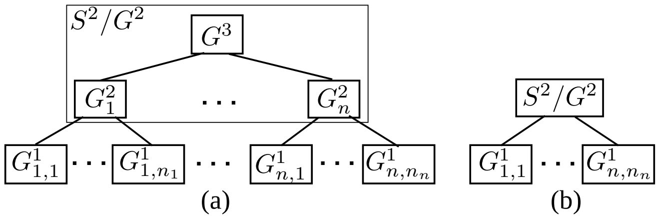

Fig. 4. Hierarchical and modular multi-level architecture

$L_{\mathrm{m}}(G_i^2) = L_{\mathrm{m}}(S_i^2/H_i^2) = \kappa_{L(H_i^2)}(L_{\mathrm{m}}(H_i^2||K_i^2))$ for specifications $K_i^2 \subseteq (\Sigma_i^2)^*$, $i = 1,\ldots,n$, and implemented as $L_{\mathrm{m}}(S_i^1/G_i^1) = L_{\mathrm{m}}(G_i^2)||L_{\mathrm{m}}(G_i^1)$ in the first level. Likewise, we have $L_{\mathrm{m}}(G^3) = L_{\mathrm{m}}(S^3/H^3) = \kappa_{L(H^3)}(L_{\mathrm{m}}(H^3)||K^3)$ for $K^3 \in (\Sigma^3)^*$ and $L_{\mathrm{m}}(S^2/G^2) = L_{\mathrm{m}}(G^3)||L_{\mathrm{m}}(G^2)$ in the second level (see Fig. 4). The overall supervisor is $L_{\mathrm{m}}(S/H) = L_{\mathrm{m}}(G^3)||L_{\mathrm{m}}(G^2)||L_{\mathrm{m}}(G^1)$.

(iv) *nonblocking control*: we require that the low-level control is nonblocking, i.e., $\overline{L_{\mathrm{m}}(S/H)} = L(S/H)$.

(v) *abstraction condition*: we require that all natural projections p_i^2 and $p_{i,k}^1$, $i = 1,\ldots,n$, $k = 1,\ldots,n_i$ are $L(G_i^2)$-observers for $L(G_i^2)$ and $L(G_{i,k}^1)$-observers for $L(G_{i,k}^1)$, respectively.

Applying the conditions in Theorem 4 to each hierarchical abstraction in the 3-level hierarchy according to Definition 8, we can state the main result of this paper.

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 $s^{\mathrm{hi}} \in L(G^3)$, p_i^2 is lcc for $L_{\mathrm{loc,i}}^2(s^{\mathrm{hi}})$, and for all $s^{\mathrm{hi}} \in L(G_i^2)$, $p_{i,k}^1$ is lcc for $L_{\mathrm{loc,i,k}}^1(s^{\mathrm{hi}})$. Furthermore let $H_{i,k}^1$ and $H_{j,l}^1$ be mutually controllable for all $i,j = 1,\ldots,n$ and $k = 1,\ldots,n_i$, $l = 1,\ldots,n_j$ such that $i \neq j$ or $k \neq l$, and H_i^2, H_j^2 be mutually controllable for all $i,j = 1,\ldots,n$, $i \neq j$. Then S is maximally permissive.

Proof: Let $K^2 := L_{\mathrm{m}}(H^2)||K^3||K_1^2||\cdots||K_n^2$ and $K^1 := L_{\mathrm{m}}(H^1)||K^2||K_{1,1}^1||\cdots||K_{n,n_n}^1$. Appealing to Theorem 4, we know that (i) $L_{\mathrm{m}}(S^2/G^2) = \kappa_{L(H^2)}(K^2) = L_{\mathrm{m}}(G^3)||L_{\mathrm{m}}(G^2)$ (situation in Fig. 4 (a)) and (ii) $\kappa_{L(H^1)}(K^1) = \kappa_{L(H^2)}(K^2)||L_{\mathrm{m}}(G^1) = L_{\mathrm{m}}(S_2/G_2)||L_{\mathrm{m}}(G^1)$ (situation in Fig. 4 (b)). Combining (i) and (ii), we arrive at $\kappa_{L(H^1)}(K^1) = L_{\mathrm{m}}(G^3)||L_{\mathrm{m}}(G^2)||L_{\mathrm{m}}(G^1)$. ■

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

[1] K. C. Wong and W. M. Wonham, "Hierarchical control of discrete-event systems," *Discrete Event Dynamic Systems: Theory and Applications*, 1996.

[2] K. Schmidt, S. Perk, and T. Moor, "Nonblocking hierarchical control of decentralized systems," *IFAC World Congress, Prague*, 2005.

[3] K. Schmidt. (2005) Hierarchical control of decentralized discrete event systems: Theory and application. Phd-thesis, Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg. [Online]. Available: http://www.rt.e-technik.uni-erlangen.de/FGdes/dissertation2005_schmidt.pdf

[4] L. Feng and W. M. Wonham, "Computationally efficient supervisor design: Abstraction and modularity," *Workshop on Discrete Event Systems*, 2006.

[5] R. Hill and D. Tilbury, "Modular supervisory control of discrete-event systems with abstraction and incremental hierarchical construction," *Workshop on Discrete Event Systems*, 2006.

[6] K. Schmidt, H. Marchand, and B. Gaudin, "Modular and decentralized supervisory control of concurrent discrete event systems using reduced system models," *Workshop on Discrete Event Systems (WODES), Ann Arbor, USA*, 2006.

[7] K. Schmidt, M. de Queiroz, and J. Cury, "Hierarchical and decentralized multitasking control of discrete event systems," *Conference on Decision and Control*, 2007.

[8] S.-H. Lee and K. C. Wong, "Structural decentralised control of concurrent DES," *European Journal of Control*, vol. 35, pp. 1125–1134, October 2002.

[9] C. G. Cassandras and S. Lafortune, "Introduction to discrete event systems," *Kluwer*, 1999.

[10] W. M. Wonham, "Notes on control of discrete event systems," *Department of Electrical Engineering, University of Toronto*, 2004.

[11] H. Zhong and W. M. Wonham, "On the consistency of hierarchical supervision in discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 35, pp. 1125–1134, October 1990.

[12] J.-C. Fernandez, "An implementation of an efficient algorithm for bisimulation equivalence," *Science of Computer Programming*, vol. 13, pp. 219–236, 1990.

[13] K. C. Wong and W. M. Wonham, "On the computation of observers in discrete-event systems," *Discrete Event Dynamic Systems*, vol. 14, no. 1, pp. 55–107, 2004.

[14] K. Schmidt and T. Moor, "Computation of marked string accepting observers for discrete event systems," *Workshop on Discrete Event Systems (WODES), Ann Arbor, USA*, 2006.

[15] T. Moor, K. Schmidt, and S. Perk, "libFAUDES – an open source C++ library for discrete event systems," *Workshop on Discrete Event Systems*, 2008.