Modular Control of Discrete-Event Systems using Similarity ★

Yingying Liu [a], Jan Komenda [b], Tomáš Masopust [c,b], Zhiwu Li [d,e]

[a]*Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China*

[b]*Institute of Mathematics of the Czech Academy of Sciences, Prague, Czechia*

[c]*Department of Computer Science, Faculty of Science, Palacky University, Olomouc, Czechia*

[d]*School of Electro-Mechanical Engineering, Xidian University, Xi'an, 710071, China*

[e]*Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau SAR, China*

Abstract

We investigate modular supervisory control of discrete-event systems composed of several groups of components, where each group consists of similar modules. Because of the similar structures of the modules, such systems can be represented as a set of (group) templates. Supervisory control can then be performed on these templates, resulting in a set of template supervisors. We propose a modular approach to construct the template supervisors based on the local computation of supremal symmetric sublanguages and on the concept of conditional decomposability. The supremal symmetric sublanguage of a decomposable language turns out to be decomposable, and can thus be computed locally. It is proven that the local supervisors of the components of a group are similar and can thus be obtained by a symmetry map from the template supervisor of the group.

Key words: Similar discrete-event systems, symmetric sublanguage, similar sublanguage, observability.

1 Introduction

In large-scale engineering systems, cf. Agarwal et al. (2019); Ding et al. (2019); Du et al. (2020), multiple agents are often divided into groups of subsystems according to their roles as in Amini et al. (2020). Within each group, the agents (such as robots or AGVs in Wu and Zhou (2007b); Luo et al. (2015)) handle similar jobs, and have therefore similar behaviors, cf. also Su and Lennartsson (2017); Wu and Zhou (2007a); Wurman et al. (2008). Such multi-agent systems are referred to as discrete-event systems (DES) with similar components, see e.g. Rohloff and Lafortune (2006); Su and Lennartsson (2017); Wu and Zhou (2007a); Wurman et al. (2008), where components in the same group have similar state transition structures.

The control problems of DES with similar components have been discussed in the literature. Eyzell and Cury (2001) investigated the symmetry of systems to reduce the complexity of supervisory control by constructing a reduced automaton. A quotienting technique has been recently presented in Basu and Kumar (2021) for simplification of non deterministic automata. Wang et al. (2019) studied blocking and deadlocking for systems with isomorphic modules. Rohloff and Lafortune (2006) explored the control and verification problems of DES with similar components, and introduced the concept of symmetry. They focused on existential results and identified the necessary and sufficient conditions for the existence of a set of similar local supervisors that enforce a given specification. These conditions include, among others, separability (also known as decomposability) and symmetry.

Since the monolithic synthesis becomes computationally expensive and infeasible for a large number of agents, modular methods have been proposed for systems with a similar structure to avoid the synthesis of a monolithic system. Jiao et al. (2017) considered systems consisting of groups of ma-

★ The paper was not presented at any conference. Corresponding author: Jan Komenda, tel. +420222090784, fax +420541218657. Partially supported by the MŠMT INTER-EXCELLENCE project LTAUSA19098, the GAČR grant 19-06175J, the National Key Research and Development Program of China (2018AAA0101700) and the National Natural Science Foundation of China (62061136004, 61803259, 61833012), Shanghai Jiao Tong University Scientific and Technological Innovation Funds and by RVO 67985840.

Email addresses: yingyingliu611@163.com (Yingying Liu), komenda@ipm.cz (Jan Komenda), tomas.masopust@upol.cz (Tomáš Masopust), zhwli@xidian.edu.cn (Zhiwu Li).

chines having an isomorphic structure and extracted the control information with the help of the state tree structures. Liu et al. (2019) investigated modular control of DES with similar components under some restrictive assumptions, such as the agents do not share events and all events are observable.

Our contribution. In this paper, we use the symmetry map of Rohloff and Lafortune (2006), which is better suited for the case, where agents share some (global) events and for the case of partial observation of events. Our work extends the results of Rohloff and Lafortune (2006) by dropping the assumptions that the specification is symmetric, decomposable, controllable, and normal. If the specification language fails to be symmetric, we show that the supremal symmetric sublanguage always exists, and we show how to compute it in a modular way (Theorem 7). We use a more general concept of conditional decomposability, which relaxes the assumption that the given specification is decomposable according to the alphabets of local agents. Concerning controllability and normality, we make use of supremal controllable and relatively observable sublanguages, and of the fact that supremal controllable and relatively observable sublanguages of a set of similar (local) specifications remain similar. This observation allows us to base our template supervisors on these notions. Moreover, Rohloff and Lafortune (2006) assume that all local agents are similar, that is, they considered only a single group of similar agents. We consider a modular system that consists of several groups of similar agents, that is, the agents from different groups need not be similar. The relaxation of these four assumptions and the similarity of all local plants lead to Theorem 10 where a *template supervisor* is designed for each group. Local supervisors of the subsystems within a group are obtained from the template supervisor of the group with the help of a *symmetry map* that maps the events of the template supervisor to the relevant global and private events of each subsystem.

2 Preliminaries

A generator $\mathbf{G} = (Z, \Sigma, \delta, z_0, Z_m)$ is a quintuple, where Σ is a finite set of events, Z is a finite set of states, $z_0 \in Z$ is an initial state, $Z_m \subseteq Z$ is a set of marker states, and $\delta\colon Z \times \Sigma \to Z$ is a (partial) transition function. As usual, δ can be extended to $\delta\colon Z \times \Sigma^* \to Z$, where Σ^* is the set of all finite-length *strings*, including the empty string ε. The *closed behavior* of $\mathbf{G}$ is the language $L(\mathbf{G}) = \{s \in \Sigma^* \mid \delta(z_0, s) \in Z\}$, and the *marked behavior* of $\mathbf{G}$ is the set $L_m(\mathbf{G}) = \{s \in L(\mathbf{G}) \mid \delta(z_0, s) \in Z_m\}$. The length of a string s is denoted by $|s|$. The *prefix closure* of a language L is the set of its prefixes. For a natural number n, let $[1, n] = \{1, \ldots, n\}$.

Let Σ be partitioned into the set of controllable events Σ_c and the set of uncontrollable events $\Sigma_u = \Sigma \setminus \Sigma_c$. A language $K \subseteq \Sigma^*$ is *controllable* with respect to $L(\mathbf{G})$ and Σ_u if $\overline{K}\Sigma_u \cap L(\mathbf{G}) \subseteq \overline{K}$. Let $\mathcal{C}(K, L(\mathbf{G})) = \{E \subseteq K \mid \overline{E}\Sigma_u \cap L(\mathbf{G}) \subseteq \overline{E}\}$ denote the set of all controllable sublanguages of K, and $\sup \mathcal{C}(K, L(\mathbf{G})) = \cup \{E \mid E \in \mathcal{C}(K, L(\mathbf{G}))\}$ its supremum.

To model partial observation, let Σ be further partitioned into the set of observable events Σ_o and the set of unobservable events $\Sigma_{uo} = \Sigma \setminus \Sigma_o$. Let $Q\colon \Sigma^* \to \Sigma_o^*$ denote the *projection* erasing all events from Σ_{uo} Cassandras and Lafortune (2008). Let $K \subseteq C \subseteq L_m(\mathbf{G})$ be languages. Then K is relatively observable with respect to $\overline{C}$, $\mathbf{G}$, and Q, or $\overline{C}$*-observable*, provided that, for every $s, s' \in \Sigma^*$ and every $\sigma \in \Sigma$, if $s\sigma \in \overline{K}$, $s' \in \overline{C}$, $s'\sigma \in L(\mathbf{G})$, and $Q(s) = Q(s')$, then $s'\sigma \in \overline{K}$ Cai et al. (2015). Let $\mathcal{O}(K, L(\mathbf{G})) = \{E \subseteq K \mid E$ is $\overline{K}$-observable with respect to $\mathbf{G}$ and Q$\}$ denote the set of all $\overline{K}$-observable sublanguages of K, and $\sup \mathcal{O}(K, L(\mathbf{G})) = \cup\{E \mid E \in \mathcal{O}(K, L(\mathbf{G}))\}$ its supremum.

Let $\mathcal{CO}(K, L(\mathbf{G})) = \mathcal{C}(K, L(\mathbf{G})) \cap \mathcal{O}(K, L(\mathbf{G}))$ denote the set of all sublanguages that are controllable with respect to $L(\mathbf{G})$ and Σ_u, and $\overline{K}$-observable with respect to $\mathbf{G}$ and Q. Then $\mathcal{CO}(K, L(\mathbf{G}))$ has supremum $\sup \mathcal{CO}(K, L(\mathbf{G}))$.

A language $K \subseteq \Sigma^*$ is *decomposable* with respect to alphabets $\Sigma_1, \ldots, \Sigma_n$ if $K = \|_{i=1}^n P_i(K)$, where $P_i\colon \Sigma^* \to \Sigma_i^*$ is the projection and $\|$ is the synchronous product, defined as follows. For local languages $L_i \subseteq \Sigma_i^*$, $\|_{i=1}^n L_i = \cap_{i=1}^n P_i^{-1}(L_i) \subseteq \Sigma^*$, where $\Sigma = \cup_{i=1}^n \Sigma_i$ is the global alphabet. For automata definition and more details see the literature Wonham and Cai (2019).

3 Modular control of DES with similar components

We investigate modular supervisory control for DES with $\ell \geq 1$ groups of similar agents, $\{\mathcal{G}_1, \ldots, \mathcal{G}_\ell\}$, i.e. for $i = 1, \ldots, \ell$, $\mathcal{G}_i = \{\mathbf{G}_{i1}, \ldots, \mathbf{G}_{in_i}\}$, where $\mathbf{G}_{ij}$, $j = 1, \ldots, n_i$, are similar as defined below. For all i, $\mathbf{G}_{ij}$ are over Σ_{ij} of the same cardinality that are decomposed as $\Sigma_{ij} = \Sigma_{pij} \cup \Sigma_{gi}$, where Σ_{gi} is the global (shared) alphabet of the group $\mathcal{G}_i$, and Σ_{pij} is the private alphabet of $\mathbf{G}_{ij}$. Since Σ_{pij} are of the same cardinality for the group $\mathcal{G}_i$, we denote the "rewriting" bijection by $\psi^i_{jj'}\colon \Sigma_{pij} \to \Sigma_{pij'}$. The set of all events in the system is denoted by $\Sigma = \cup_{i=1}\Sigma_i$, where $\Sigma_i = \cup_{j=1}^{n_i}\Sigma_{ij}$ is the alphabet of group $\mathcal{G}_i$. The group symmetry map $\Psi^i_{jj'}\colon \Sigma \to \Sigma$ for group $\mathcal{G}_i$ is defined as

$$\Psi^i_{jj'}(\sigma) = \begin{cases} \psi^i_{jj'}(\sigma) & \text{if } \sigma \in \Sigma_{pij} \cup \Sigma_{pij'} \\ \sigma & \text{if } \sigma \in \Sigma \setminus (\Sigma_{pij} \cup \Sigma_{pij'}) \end{cases} \tag{1}$$

Note that $\Psi^i_{jj'}$ interchanges the private events of agents G_{ij} and $G_{ij'}$ in the i-th group, while all the other events are kept unchanged. For instance, if $i = 1$ and $\Sigma_{1j} = \{a_j, a\}$, $j = 1, \ldots, n_1$, with a being the shared event, then $\Psi^1_{jj'}(a_j) = a_{j'}$, $\Psi^1_{jj'}(a_{j'}) = a_j$, $\Psi^1_{jj'}(a) = a$ and $\Psi^1_{jj'}(a_k) = a_k$ for all $k \notin \{j, j'\}$. We extend $\Psi^i_{jj'}$ to $\Psi^i_{jj'}\colon \Sigma^* \to \Sigma^*$ in a usual way, i.e. $\Psi^i_{jj'}(\varepsilon) = \varepsilon$ and $\Psi^i_{jj'}(wa) = \Psi^i_{jj'}(w)\Psi^i_{jj'}(a)$ for $a \in \Sigma$ and $w \in \Sigma^*$. Moreover, $\Psi^i_{jj'}$ is extended to languages by $\Psi^i_{jj'}(L) = \{\Psi^i_{jj'}(w) \in \Sigma^* | w \in L\}$, $L \subseteq \Sigma^*$.

For any $j, j' \in [1, n_i]$, $\Psi^i_{jj'} = \Psi^i_{j'j}$, $(\Psi^i_{jj'})^{-1} = \Psi^i_{j'j} = \Psi^i_{jj'}$, and $\Psi^i_{jj} = \Psi^i_{jj'} \circ \Psi^i_{j'j}$ is the identity. Moreover, $\Psi^i_{jj'}$ is monotone, i.e. for $K \subseteq K'$, $\Psi^i_{jj'}(K) \subseteq \Psi^i_{jj'}(K')$.

A group of agents $\mathcal{G}_i$ is said to be *similar* under $\Psi^i_{jj'}$ if

$$(\forall j, j' \in [1, n_i])\ L(\mathbf{G}_{ij}) = \Psi^i_{jj'}(L(\mathbf{G}_{ij'}))\,.$$

Note that this means that every agent can be obtained from another by interchanging their corresponding private events. The following property is often used. If for all $j, j' \in [1, n_i]$, $\Psi^i_{jj'}(L(\mathbf{G}_{ij}) \subseteq L(\mathbf{G}_{ij'})$, then the agents in $\mathcal{G}_i$ are similar. Indeed, the other inclusion follows by applying $\Psi^i_{jj'}$ on both sides from monotonicity and

$$\Psi^i_{jj} = \Psi^i_{jj'} \circ \Psi^i_{j'j} \text{ is the identity.} \tag{2}$$

We assume that each group of agents $\mathcal{G}_i$ is composed of similar generators under $\Psi^i_{jj'}$, that is, $\mathcal{G}_i = \{\mathbf{G}_{i1}, \ldots, \mathbf{G}_{in_i}\}$, where $L(\mathbf{G}_{ij}) = \Psi^i_{jj'}(L(\mathbf{G}_{ij'}))$, for all $j, j' \in [1, n_i]$.

The plant $\mathbf{G}$ is the *synchronous product* of all agents:

$$\mathbf{G} = \|_{i=1}^{\ell} \mathbf{G}_i, \quad \text{where } \mathbf{G}_i = \|_{j=1}^{n_i} \mathbf{G}_{ij}. \tag{3}$$

Since $\mathbf{G}$ is decomposed in a top-down way, we conditionally decompose a global specification K in the top-down way as well. Let $\cup_{i \neq j \in [1,\ell]}(\Sigma_i \cap \Sigma_j) \subseteq \Sigma_k \subseteq \Sigma$ and

$$\Sigma_{gi} \subseteq \Sigma_{k_i} \subseteq \Sigma_i\,,$$

where Σ_{gi} are low-level global alphabets and Σ_{k_i} are low-level coordinator alphabets. A language K is *two-level conditionally decomposable* with respect to alphabets $\Sigma_1, \ldots, \Sigma_\ell$, a high-level coordinator alphabet Σ_k, and the low-level coordinator alphabets $\Sigma_{k_1}, \ldots, \Sigma_{k_\ell}$ if

$$K = \|_{i=1}^{\ell} P_{i+k}(K), \text{ and} \tag{4}$$

$$P_{i+k}(K) = \|_{j=1}^{n_i} P_{ij+k_i+k}(K)\,, \tag{5}$$

where $P_{i+k} \colon \Sigma^* \to (\Sigma_i \cup \Sigma_k)^* = (\cup_{j=1}^{n_i} \Sigma_{ij} \cup \Sigma_k)^*$ and $P_{ij+k_i+k} \colon \Sigma^* \to (\Sigma_{ij} \cup \Sigma_{k_i} \cup \Sigma_k)^*$, $i = 1, \ldots, \ell$, $j = 1, \ldots, n_i$. Here we consider the coordinated group alphabets $\Sigma_i \cup \Sigma_k$, where Σ_k is the top coordinator alphabet. Note that the coordinator alphabet Σ_{k_i} of the group coordinator $\mathbf{G}_{k_i}$ contains all shared events of the group, i.e. $\Sigma_{k_i} \supseteq \Sigma_{gi}$. The two-level conditional decomposability extends conditional decomposability to large plants structured into groups and enables to consider a non-decomposable specification as decomposable over extended local alphabets (enriched by coordinator events).

We say that a two-level conditionally decomposable language K is *symmetric by group* (according to the alphabets inside groups) if for every group $i = 1, \ldots, \ell$ we have

$$(\forall j, j' \in [1, n_i])\ P_{ij+k_i+k}(K) = \Psi^i_{jj'}(P_{ij'+k_i+k}(K))\,,$$

that is, the languages $P_{i1+k_i+k}(K), \ldots, P_{in_i+k_i+k}(K)$ are similar under $\Psi^i_{jj'}$. We point out that the specification $K \subseteq L_m(\mathbf{G})$ does not need to be symmetric and decomposable. If the specification is not decomposable, then the two-level conditional decomposability can be used to decompose the specification according to the two-level (group) structure of subsystems.

Now we recall the multilevel coordination control synthesis Komenda et al. (2013) for groups of local subsystems, under the assumption that K is two-level conditionally decomposable. The local supervisors $\mathbf{S}_{i1}, \ldots, \mathbf{S}_{in_i}$ for generators extended by the coordinator events are computed by

$$L_m(\mathbf{S}_{ij}) = \sup \mathcal{CO}(P_{ij+k_i+k}(K), L(\mathbf{G}_{ij} \| \mathbf{G}_{k_i} \| \mathbf{G}_k)) \tag{6}$$

where $\mathbf{G}_k = \|_{i=1}^{\ell} P_k(\mathbf{G}_i)$ is the high-level coordinator, $\mathbf{G}_{k_i} = \|_{j=1}^{n_i} P_{k_i}(\mathbf{G}_{ij})$ are the low-level (group) coordinators for $i = 1, \ldots, \ell$, $P_k : \Sigma^* \to \Sigma_k^*$ and $P_{k_i} : \Sigma_i^* \to \Sigma_{k_i}^*$. Let us write equation (6) as

$$S_{ij} = \sup \mathcal{CO}(P_{ij+k_i+k}(K), L_{ij+k_i+k}), \tag{7}$$

where $S_{ij} = L_m(\mathbf{S}_{ij})$ and $L_{ij+k_i+k} = L(\mathbf{G}_{ij}) \| L(\mathbf{G}_{k_i}) \| L(\mathbf{G}_k)$.

Since the underlying plants are given by synchronous products of the plants with the group coordinator and the top coordinator, the products $\mathbf{G}_{ij} \| \mathbf{G}_k \| \mathbf{G}_{k_i}$ play the role of the local plants $\mathbf{G}_{ij}$. If the specification is not symmetric, then we can compute the supremal symmetric sublanguage of K by Theorem 7.

Because global coordinator events from Σ_k might belong to different group alphabets, we need $\Psi^i_{jj'}$ to be defined on the whole alphabet Σ. Indeed, in the group specification $P_{i+k}(K)$ there can be high-level coordinator events from $\Sigma \setminus \Sigma_i$, and hence in (1) we consider the whole alphabet Σ and not just Σ_i. If local specification languages of the group $\mathcal{G}_i$, namely $P_{ij+k_i+k}(K), j = 1, \ldots, n_i$ are not similar with respect to $\Psi^i_{jj'}$, we first compute the set of similar sublanguages for this (coordinated) group, which needs the transitivity property of the group symmetry map.

Note that transitivity of the group symmetry map does not hold, i.e., $\Psi^i_{jj'}\Psi^i_{j'\ell} \neq \Psi^i_{j\ell}$. The problem is that in conditional decompositions the local alphabets extended by coordinator events are in general larger than the private alphabets, while we insist on keeping the symmetry for the originally given distribution of local alphabets $\Sigma_{ij} = \Sigma_{pij} \cup \Sigma_{gi}$ into private alphabets and global alphabets. To make sure that the transitivity of the symmetric map holds for two-level conditional decomposable languages, we give the following lemma with a condition that guarantees transitivity.

Lemma 1 *Consider the local specification languages* $K_{il} \subseteq (\Sigma_{il} \cup \Sigma_{k_i} \cup \Sigma_k)^*$ *of the group* $\mathcal{G}_i$ *for some* $l \in \{1, \dots n_i\}$. *Assume that for all* $j, j' \in \{1, \dots n_i\}$ *such that* $j \neq l$ *and* $j' \neq l$ *we have* $\Psi^i_{jj'}(K_{il}) = K_{il}$. *Then* $\Psi^i_{jj'}\Psi^i_{j'l}(K_{il}) = \Psi^i_{jl}(K_{il})$.

PROOF. We first show that the assumption implies that for all $j, j' \in [1, n_i]$ such that $j \neq l$ and $j' \neq l$ we have $\Psi^i_{jj'}\Psi^i_{j'l}(s) = \Psi^i_{jl}\Psi^i_{jj'}(s)$. We apply Ψ^i_{jl} on both sides of $\Psi^i_{jj'}(K_{il}) = K_{il}$ and for all $j, j' \in \{1, \dots n_i\}$ such that $j \neq l$ and $j' \neq l$ we obtain $\Psi^i_{jl}\Psi^i_{jj'}(K_{il}) = \Psi^i_{jl}(K_{il})$. Now we need to show that $\Psi^i_{jj'}\Psi^i_{j'l}(K_{il}) = \Psi^i_{jl}\Psi^i_{jj'}(K_{il})$ for all $j, j' \in [1, n_i]$ such that $j \neq l$ and $j' \neq l$. We use induction to prove it. Assume $s \in K_{il}$. We will show that $\Psi^i_{jj'}\Psi^i_{j'l}(s) = \Psi^i_{jl}\Psi^i_{jj'}(s)$.

The base step is immediate, for $s = \varepsilon$ the claim holds true. Assume that the claim holds for all strings s of length $n > 0$, i.e. $\Psi^i_{jj'}\Psi^i_{j'l}(s) = \Psi^i_{jl}\Psi^i_{jj'}(s)$. Now we show that for every $a \in \Sigma$ such that $sa \in K_{il}$ we have $\Psi^i_{jj'}\Psi^i_{j'l}(sa) = \Psi^i_{jl}\Psi^i_{jj'}(sa)$. Since $\Psi^i_{jj'}\Psi^i_{j'l}(sa) = \Psi^i_{jj'}\Psi^i_{j'l}(s)\Psi^i_{jj'}\Psi^i_{j'l}(a) = \Psi^i_{jl}\Psi^i_{jj'}(s)\Psi^i_{jj'}\Psi^i_{j'l}(a)$, we prove that for every $a \in \Sigma$, $\Psi^i_{jj'}\Psi^i_{j'l}(a) = \Psi^i_{jl}\Psi^i_{jj'}(a)$. We distinguish the following cases:

If $a = a_j \in \Sigma_{pij}$, then since $sa \in K_{il}$, it follows from the assumption $\Psi^i_{jj'}(K_{il}) = K_{il}$ for all $j, j' \neq l$ that $\Psi^i_{j'l}(a_j) = a_j$ in case $j' \neq j$. Therefore, $\Psi^i_{jj'}\Psi^i_{j'l}(a_j) = \Psi^i_{jj'}(a_j) = a_{j'}$ and $\Psi^i_{jl}\Psi^i_{jj'}(a_j) = \Psi^i_{jl}(a_{j'}) = a_{j'}$, as well, where $a_{j'} = \psi^i_{jj'} \in \Sigma_{pij'}$ and $\Psi^i_{jl}(a_{j'}) = a_{j'}$ follows again from our assumption and $j' \neq l$. If $j' = j$ and $j' \neq l$ then we have $\Psi^i_{j'l}(a_j) = a_l$, but also $\Psi^i_{jj'}\Psi^i_{j'l}(a_j) = \Psi^i_{jj'}(a_l) = a_l$ and $\Psi^i_{jl}\Psi^i_{jj'}(a_j) = \Psi^i_{jl}(a_j) = a_l$.

If $a = a_l \in \Sigma_{pil}$, then since $sa \in K_{il}$, it follows from the assumption that $\Psi^i_{jj'}(a_l) = a_l$. Therefore, $\Psi^i_{jj'}\Psi^i_{j'l}(a_l) = \Psi^i_{jj'}(a_{j'}) = a_j$ and $\Psi^i_{jl}\Psi^i_{jj'}(a_l) = \Psi^i_{jl}(a_l) = a_j$, where $a_j = \psi^i_{jl}(a) \in \Sigma_{pij}$. In case $j' = j$ nothing is changed in the above case, both sides equal a_j;

If $a = a_{j'} \in \Sigma_{pij'}$, then $\Psi^i_{jj'}\Psi^i_{j'l}(a_{j'}) = \Psi^i_{jj'}(a_l) = a_l$ and $\Psi^i_{jl}\Psi^i_{jj'}(a_{j'}) = \Psi^i_{jl}(a_j) = a_l$, where $a_l = \psi^i_{jl}(a_j) \in \Sigma_{pil}$. In case $j' = j$ both sides equal a_l.

If $a \in \Sigma \setminus (\Sigma_{pij} \cup \Sigma_{pij'} \cup \Sigma_{pil})$, then $\Psi^i_{jj'}\Psi^i_{j'l}(a) = \Psi^i_{jj'}(a) = a = \Psi^i_{jl}\Psi^i_{jj'}(a)$, because a is not changed by the definition of the group symmetry map. We thus prove that for all $j, j' \in [1, n_i]$ such that $j \neq l$ and $j' \neq l$ we have $\Psi^i_{jj'}\Psi^i_{j'l}(s) = \Psi^i_{jl}\Psi^i_{jj'}(s)$.

The cases $l = j$ and $l = j'$ do not require any assumption. If $l = j$ then $\Psi^i_{jj'}\Psi^i_{j'l}(K_{il}) = \Psi^i_{jj'}\Psi^i_{j'j}(K_{ij}) = \Psi^i_{jj}(K_{ij}) = \Psi^i_{jl}(K_{il})$. Similarly, If $l = j'$ then $\Psi^i_{jj'}\Psi^i_{j'l}(K_{il}) = \Psi^i_{jj'}\Psi^i_{j'j'}(K_{ij'}) = \Psi^i_{jj'}(K_{ij'}) = \Psi^i_{jl}(K_{il})$. That is, $s \in \cap_{l=1}^{n_i}\Psi^i_{jl}(K_{il})$. This completes the proof of transitivity. □

The assumption in Lemma 1 requires that coordinated events from $\Sigma_{k_i} \cup \Sigma_k$ are added to local alphabets Σ_{il} such that do not alter symmetry. It might look at first sight that the assumption is not restrictive at all and it always holds true, but remember that $K_{il} \subseteq (\Sigma_{il} \cup \Sigma_{k_i} \cup \Sigma_k)^*$, because $K_{il} = P_{il+k_i+k}(K)$ contains in general events from Σ_{ij} for some $j \neq l$. The meaning of our assumption is that after coordinator events are added to the local alphabets (as proposed in the algorithms for computing coordinator alphabets, see Komenda et al. (2012, 2013)), the local specifications remain similar.

Now we show how to obtain sets of similar specification languages in the groups.

Proposition 2 *Let us consider the framework introduced in (4) and (5), let* $K = \|_{i=1}^{\ell} P_{i+k}(K)$ *be a two-level decomposable language with* $P_{i+k}(K) = \|_{j=1}^{n_i} P_{ij+k_i+k}(K)$ *and denote* $K_{il} = P_{il+k+k_i}(K)$. *Let* $K'_{ij} = \cap_{j'=1}^{n}\Psi^i_{jj'}(K_{ij'}) = \cap_{j'=1}^{n}\Psi^i_{jj'}(P_{ij'+k_i+k}(K))$. *If* $\Psi^i_{jj'}(K_{il}) = K_{il}$ *for all* $j, j' \in \{1, \dots n_i\}$ *such that* $j \neq l$ *and* $j' \neq l$, *then for all* $j, j' \in [1, n_i]$ *we have* $K'_{ij} = \Psi^i_{jj'}(K'_{ij'})$, *i.e.*

$$\cap_{l=1}^{n_i}\Psi^i_{jl}(K_{il}) = \Psi^i_{jj'}\left(\cap_{l=1}^{n_i}\Psi^i_{lj'}(K_{il})\right) .$$

Stated in words, the new local specification languages $K'_{i1}, \dots, K'_{in}$ *in group* $\mathcal{G}_i$ *are similar.*

PROOF. Let $s \in \Psi^i_{jj'}\left(\cap_{l=1}^{n_i}\Psi^i_{lj'}(K_{il})\right)$. Then, there exists a string $s' \in \cap_{l=1}^{n_i}\Psi^i_{lj'}(K_{il})$ such that $\Psi^i_{jj'}(s') = s$. We have $s' \in \cap_{l=1}^{n_i}\Psi^i_{lj'}(K_{il}) = \cap_{l=1}^{n_i}\Psi^i_{j'l}(K_{il})$, thus $s' \in \Psi^i_{j'l}(K_{il})$ for $l = 1, \dots, n_i$. Consequently, $s = \Psi^i_{jj'}(s') \in \Psi^i_{jj'}\Psi^i_{j'l}(K_{il}) = \Psi^i_{jl}(K_{il})$, where Lemma 1 is used in the last equality. Thus, $\Psi^i_{jj'}(\cap_{l=1}^{n_i}\Psi^i_{lj'}(K_{il})) \subseteq \cap_{l=1}^{n_i}\Psi^i_{jl}(K_{il})$. Since for similarity one inclusion is enough, see (2), K'_{ij} for all $j, j' \in [1, n_i]$ are similar. □

To prove the equivalence relationship between symmetry of a language and similarity of its local projections, we need Lemma 3 and Lemma 4. The following lemma relates the local projections to symmetry maps.

Lemma 3 *Let* $s \in \Sigma^*$. *If* $s = \Psi^i_{jj'}(s')$, *then* $\Psi^i_{jj'}(P_j(s)) = P_{j'}(s')$ *and* $\Psi^i_{jj'}(P_{j'}(s)) = P_j(s')$. *Moreover,* $s = \Psi^i_{jj'}(s')$ *implies that* $P_l(s) = P_l(s')$ *for* $l \neq j$ *and* $l \neq j'$.

PROOF. Note that $s = \Psi^i_{jj'}(s')$ means that s and s' are of the same length. We use induction on the length of s. The base step is immediate, because for $s = s' = \varepsilon$ the claim holds true. Assume that the claim holds for strings of length $n > 0$, and consider two strings $s = ta$ and $s' = t'b$ for which $s = \Psi^i_{jj'}(s')$. Then, $a = \Psi^i_{jj'}(b)$. According to the definition of Ψ_{ij}, we have the following cases: (1) if $b \in \Sigma_{gi} \cup (\Sigma \setminus (\Sigma_{ij} \cup \Sigma_{ij'}))$, then $b = a$ and the claim holds by induction; (2) if $b \in \Sigma_{ij}$ and $a \in \Sigma_{ij'}$, then $\Psi^i_{jj'}(P_j(ta)) = P_{j'}(t'b)$ and $\Psi^i_{jj'}(P_{j'}(ta)) = P_j(t'b)$, which follows from the induction hypothesis and the fact that both the projections and the symmetry maps are catenative; (3) the case where $b \in \Sigma_{ij'}$ and $a \in \Sigma_{ij}$ is analogous to (2). The last claim that $s = \Psi^i_{jj'}(s')$ implies $P_l(s) = P_l(s')$, for $l \neq j$ and $l \neq j'$, follows directly from the definition of $\Psi^i_{jj'}$. □

We also need in the proof of Proposition 5 a lemma about distributivity of $\Psi_{jj'}$ with the synchronous product.

Lemma 4 *Let $K_l \subseteq \Sigma^*_{il}$, for $l = 1, \ldots, n$. Then $\Psi^i_{jj'}(\|_{l=1}^n K_l) = \|_{l=1}^n \Psi^i_{jj'}(K_l)$.*

PROOF. Let $s \in \Psi^i_{jj'}(\|_{l=1}^n K_l)$. Then there is $s' \in \|_{l=1}^n K_l$ such that $s = \Psi^i_{jj'}(s')$. We show that $s \in \|_{l=1}^n \Psi^i_{jj'}(K_l)$. By definition of $\Psi^i_{jj'}$, we distinguish three cases.

(1) If $j \neq l \neq j'$, then $\Psi^i_{jj'}(K_l) = K_l \subseteq \Sigma^*_{il}$. Hence, we show that $P_l(s) \in \Psi^i_{jj'}(K_l) = K_l$. Since $s = \Psi^i_{jj'}(s')$, we have that $P_l(s) = P_l(s') \in P_l(\|_{r=1}^n K_r) \subseteq K_l$.
(2) If $l = j$, then $\Psi^i_{jj'}(K_j) \subseteq \Sigma^*_{ij'}$. Hence, we show that $P_{ij'}(s) \in \Psi^i_{jj'}(K_j)$. However, from Lemma 3 we have that $s = \Psi^i_{jj'}(s')$ implies $P_{j'}(s) = \Psi^i_{jj'}(P_j(s')) \in \Psi^i_{jj'}(P_j(\|_{l=1}^n K_l)) \subseteq \Psi^i_{jj'}(K_j)$.
(3) The case $l = j'$ is symmetric to the previous case.

Let $s \in \|_{l=1}^n \Psi^i_{jj'}(K_l)$. We show that $s \in \Psi^i_{jj'}(\|_{l=1}^n K_l)$. We again distinguish three cases.

(1) If $j \neq l \neq j'$, then $\Psi^i_{jj'}(K_l) = K_l \subseteq \Sigma^*_{il}$, i.e $s \in \|_{l=1}^n \Psi^i_{jj'}(K_l)$ implies that $P_l(s) \in \Psi^i_{jj'}(K_l) = K_l$.
(2) If $l = j$ then $\Psi^i_{jj'}(K_j) \subseteq \Sigma^*_{ij'}$. Hence, $s \in \|_{l=1}^n \Psi^i_{jj'}(K_l)$ implies that $P_{j'}(s) \in \Psi^i_{jj'}(K_j)$. This means that $P_{j'}(s) = \Psi^i_{jj'}(s'_j)$ for some $s'_j \in K_j$.
(3) Case $l = j'$ is symmetric to $l = j$.

We need to show that $s = \Psi^i_{jj'}(s')$ for some $s' \in \|_{l=1}^n K_l$, i.e. $P_l(s') \in K_l$ for all $l = 1, \ldots, n$. For $s' = \Psi^i_{jj'}(s)$, $\Psi^i_{jj'}(s') = \Psi^i_{jj'}\Psi^i_{jj'}(s) = s$, and we can see that $s' \in (\|_{l=1}^n K_l)$. Indeed, for $j \neq l \neq j'$, $P_l(s) \in K_l$, i.e., $P_l(s') = P_l(s) \in K_l$ as well. Lemma 3 and $s = \Psi^i_{jj'}(s')$ imply that $P_j(s') = \Psi^i_{jj'}P_{j'}(s) = \Psi^i_{jj'}\Psi^i_{jj'}(s'_j) = s'_j \in K_j$, and $P_{j'}(s') = \Psi^i_{jj'}P_j(s) = \Psi^i_{jj'}\Psi^i_{jj'}(s'_{j'}) = s'_{j'} \in K_{j'}$. Hence, $s' \in \|_{l=1}^n K_l$ and $s \in \Psi^i_{jj'}(\|_{l=1}^n K_l)$. □

We recall from Rohloff and Lafortune (2006) that symmetric languages over group alphabets are fixpoints of all symmetry maps. A language K is *symmetric* if, for every $j, j' \in [1, n]$, $K = \Psi_{jj'}(K)$.

Example 1 *As a running example, let $\Sigma_1 = \{a_1, c\}$, $\Sigma_2 = \{a_2, c\}$ for $K = \{\overline{a_1a_2c}, \overline{a_2a_1c}\}^*$, cf. right of figure 1. Since $\Psi_{12}(c) = c$, $\Psi_{12}(a_1) = a_2$, and $\Psi_{12}(a_2) = a_1$, we have $K = \Psi_{12}(K)$, i.e. K is symmetric.*

The relationship between symmetry of a language and similarity of its local projections is needed for investigation of existence and computation of supremal symmetric sublanguages, but it is interesting by itself that a language over global alphabet is symmetric iff its local projections form a set of similar languages.

Proposition 5 *1. Let $K = \|_{j=1}^n P_j(K)$. Then, K is symmetric iff the languages $P_1(K), \ldots, P_n(K)$ are similar.*
2. For a general decomposition $K = \|_{j=1}^n K_j$ for some $K_j \subseteq \Sigma^*_j$, $j \in [1, n]$, if the languages $K_1, \ldots, K_n$ are similar, then K is symmetric.

PROOF. 1. "⇒ " Assume that K is symmetric. We need to show that for all $i, j \in [1, n]$, $P_j(K) = \Psi_{jj'}(P_{j'}(K))$. However, Lemma 4 implies that $K = \Psi_{jj'}(K) = \|_{\ell=1}^n \Psi_{jj'}(P_\ell(K))$ for all $j, j' \in [1, n]$ is a decomposition of K. We know that $P_{j'}(K) \subseteq \Psi_{jj'}(P_j(K))$, cf. Willner and Heymann (1991). Then, $\Psi_{jj'}(P_{j'}(K)) \subseteq \Psi_{jj'}\Psi_{jj'}(P_j(K)) = P_j(K)$.

"⇐" We need to show that, for all $j, j' \in [1, n]$, if $P_j(K) = \Psi_{jj'}(P_{j'}(K))$, then K is symmetric. However, $\Psi_{jj'}(K) = \Psi_{jj'}(\|_{\ell=1}^n P_\ell(K)) = \|_{\ell=1}^n \Psi_{jj'}(P_\ell(K))$, by Lemma 4, which is equal to $\|_{\ell=1}^n P_\ell(K)$ by the similarity assumption, and hence it is equal to K.

2. The same proof as in "⇐" above can be used with the only difference that $P_j(K)$ is replaced by arbitrary local languages K_j for all $j \in [1, n]$, because Lemma 4 holds for arbitrary decompositions. □

Returning to the running example we check that local projections of K are similar languages. Indeed, $\Psi_{12}(P_1(K)) = \Psi_{12}(\{\overline{a_1c}\}^*) = \{\overline{a_2c}\}^* = P_2(K)$, which is to be expected from Proposition 5 given that we established $K = P_1(K)\|P_2(K) = \{\overline{a_1a_2c}, \overline{a_2a_1c}\}^*$ as a symmetric language.

Finally, we will show that supremal symmetric sublanguages always exist. Let $K \subseteq \Sigma_i^*$ be a group specification and

$$\mathcal{S}(K) = \{K' \subseteq K \mid K' \text{ is symmetric under the symmetric maps } \Psi^i_{jj'} \text{ for all } j, j'\} \, .$$

Theorem 6 *The set $\mathcal{S}(K)$ is nonempty and closed under arbitrary unions. In particular, $\mathcal{S}(K)$ contains a (unique) supremal element, $\sup \mathcal{S}(K)$, that is equal to*

$$\sup \mathcal{S}(K) = \cup \{K' \mid K' \in \mathcal{S}(K)\} \, .$$

PROOF. The set $\mathcal{S}(K)$ is nonempty, because the empty language is symmetric and included in $\mathcal{S}(K)$. To show that $\mathcal{S}(K)$ is closed under arbitrary unions, let I be an index set such that $K^l \in \mathcal{S}(K)$ for every $l \in I$, that is, $K^l = \Psi^i_{jj'}(K^l)$ for all $j, j' \in [1, n_i]$. We show that $M = \cup_{l \in I} K^l$ is symmetric, that is, $M = \Psi^i_{jj'}(M)$ for every $j, j' \in [1, n]$. To show that $\Psi^i_{jj'}(M) \subseteq M$, let $s \in \Psi^i_{jj'}(M)$. Then, there exists $l \in I$ and $s' \in K^l$ such that $s = \Psi^i_{jj'}(s')$, and therefore $s = \Psi^i_{jj'}(s') \in \Psi^i_{jj'}(K^l) = K^l \subseteq M$. The other inclusion is not needed cf. (2). □

It follows from Theorem 7 below that $\|_{j=1}^{n_i} K'_{ij}$ is the supremal sublanguage of $P_{i+k}(K)$ that is symmetric according to $\Psi^i_{jj'}$. For simplicity we omit in this theorem the group index i and replace conditional decomposability with standard decomposability, namely the role of $P_{i+k}(K) = \|_{j=1}^{n_i} P_{ij+k_i+k}(K)$ is played by $K = \|_{j=1}^{n} K_j$.

Theorem 7 *Let $K = \|_{j=1}^{n} K_j$ be a decomposable language. Then, the supremal symmetric sublanguage of K is equal to the synchronous product of local similar sublanguages, i.e.,*

$$\sup \mathcal{S}(K) = \|_{j=1}^{n} \left(\cap_{j'=1}^{n} \Psi_{jj'}(K_j) \right) \, . \tag{8}$$

PROOF. We note that $\|_{j=1}^{n} \left(\cap_{j'=1}^{n} \Psi_{jj'}(K_{j'}) \right)$ is symmetric. From Proposition 2, the languages

$$K'_j = \cap_{j'=1}^{n} \Psi_{jj'}(K_{j'}), \; j = 1, \ldots, n \tag{9}$$

form a similar set, and $\|_{j=1}^{n} \left(\cap_{j'=1}^{n} \Psi_{jj'}(K'_j) \right)$ is symmetric by Proposition 5. In order to simplify the proof of supremality we first show that the right hand side of (8) simplifies to $\cap_{j,j'=1}^{n} \Psi_{jj'}(K)$. It amounts to show that $\|_{j=1}^{n} \left(\cap_{j'=1}^{n} \Psi_{jj'}(K_{j'}) \right) = \cap_{j,j'=1}^{n} \Psi_{jj'}(K)$. We have $\cap_{j,j'=1}^{n} \Psi_{jj'}(K) = \cap_{j<j'} \Psi_{jj'}(K)$. Furthermore, by Lemma 4,

$$\cap_{j<j'} \Psi_{jj'}(K) = \cap_{j<j'} \Psi_{jj'}(\|_{l=1}^{n} K_l) = \cap_{j<j'} \|_{l=1}^{n} \Psi_{jj'}(K_l). \tag{10}$$

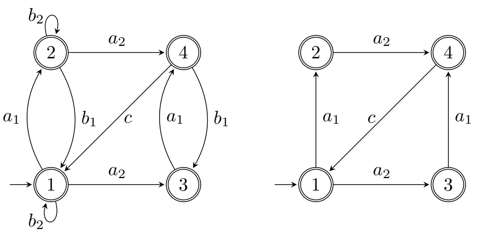

Figure 1. Generators for R (left) and $K = \sup \mathcal{S}(R)$(right).

By definition of the symmetry maps we have that

$$\Psi_{jj'}(K_l) = \begin{cases} K_l \subseteq \Sigma_l^* & \text{if } l \neq j \text{ and } l \neq j' \\ \Psi_{jj'}(K_j) \subseteq \Sigma_{j'}^* & \text{if } l = j \text{ and} \\ \Psi_{jj'}(K_{j'}) \subseteq \Sigma_j^* & \text{if } l = j' \end{cases}$$

Therefore, for any $j < j'$ we get that $\|_{l=1}^{n} \Psi_{jj'}(K_l) =$

$$K_1 \| \ldots \| K_{j-1} \| \Psi_{jj'}(K_{j'}) \| K_{j+1} \| \ldots \| \Psi_{jj'}(K_j) \| \ldots \| K_n .$$

Note that we have already put $\Psi_{jj'}(K_{j'})$, resp. $\Psi_{jj'}(K_j)$, to their corresponding places by taking into account that these languages are over alphabets Σ_j, resp. $\Sigma_{j'}$.

Since the synchronous product distributes with the language intersection we can collect in all the components the intersections (over $j, j' = 1, \ldots, n$ such that $j < j'$) of all languages over the same alphabets. By collecting the terms over the alphabet Σ_l, $l \in [1, n]$ we obtain the language $\Psi_{1l}(K_1) \cap \cdots \cap \Psi_{1l-1}(K_{l-1}) \cap K_l \cap \Psi_{ll+1}(K_{l+1}) \cap \cdots \cap \Psi_{ln}(K_n) = K'_l = K_l \cap \cap_{j' \neq l} \Psi_{lj'}(K_{j'})$. Therefore, we obtain $\cap_{j,j'=1}^{n} \Psi_{jj'}(K) = \cap_{j<j'} \|_{l=1}^{n} \Psi_{jj'}(K_l) = \|_{l=1}^{n} K'_l = \|_{l=1}^{n} \left(\cap_{j'=1}^{n} \Psi_{lj'}(K_{j'}) \right)$, where the first equality is (10).

It remains to show supremality. Let $M \subseteq K$ be such that $M = \Psi_{jj'}(M)$ for all $j, j' \in [1, n]$. By monotonicity of $\Psi_{jj'}$ we have $\Psi_{jj'}(M) \subseteq \Psi_{jj'}(K)$. Since $M = \Psi_{jj'}(M)$, we obtain that $M \subseteq \Psi_{jj'}(K)$ for all $j, j' \in [1, n]$, and hence $M \subseteq K \cap \cap_{j \neq j'} \Psi_{jj'}(K)$. □

The formula presented in Theorem 7 is only valid for languages that are decomposable according to the alphabets from the symmetry map. It is an open problem how to compute the supremal symmetric sublanguage of indecomposable languages.

Let us return once more to the running example. Consider language $R = \{\overline{a_1 b_2^* a_2 c, a_2 a_1 c, a_1 b_1, b_2^*}\}^*$ depicted on figure 1, where $b_1 \in \Sigma_1 \setminus \Sigma_2$ is another private event of the first alphabet and $b_2 \in \Sigma_2 \setminus \Sigma_1$ is the corresponding private event of Σ_2. Note that $R = P_1(R) \| P_2(R)$ is decomposable with $P_1(R) = \{\overline{a_1 c, a_1 b_1}\}^*$ and $P_2(R) = \{\overline{a_2 c, b_2}\}^*$. Therefore, according to Theorem 7 we have the following formula for the supremal symmetric sublanguage of R:

$$\sup \mathcal{S}(R) = (P_1(R) \cap \Psi_{12}(P_1(R))) \,\|\, (P_2(R) \cap \Psi_{21}(P_2(R)))\,.$$

Note that we have $\sup \mathcal{S}(R) = K$ with K from Example 1.

From now on we can assume without loss of generality that K is conditionally symmetric by group, i.e. for each group $i = 1, \ldots, \ell$ and for all local specifications $K_{ij} = P_{ij+k+k_i}(K)$, $j = 1, \ldots, n_i$, we have $K_{ij} = \Psi^i_{jj'}(K_{ij'})$, because otherwise we can compute sublanguages $K'_{ij} \subseteq K_{ij}$, $j = 1, \ldots, n_i$, as in Proposition 2 that are similar. We recall the assumption that the group of agents $\mathcal{G}_i$ is *symmetric* under $\Psi^i_{jj'}$, namely that

$$(\forall j, j' \in [1, n_i])\; L(\mathbf{G}_{ij}) = \Psi^i_{jj'}(L(\mathbf{G}_{ij'}))\,.$$

Following the classical synthesis method given in (7), the local supervisors $\mathbf{S}_{ij}$ can be designed for agents $\mathbf{G}_{ij}$ individually. The computation is, however, intractable without an upper bound on the number of agents. If local supervisors in the same group are similar then one may synthesize a template supervisor for all local supervisors in one group. The local supervisors in this group can be obtained by replacing the private events of the template by the private events of the corresponding component. To prove similarity of local supervisors in the same group, we need the following result stating that the symmetry map commutes with natural projection.

Lemma 8 *Let* $s \in \Sigma_i^* = (\cup_{j=1}^{n_i} \Sigma_{ij})^*$, $\Sigma_{ijo} \subseteq \Sigma_{ij}$ *be the observable alphabets, and let* $Q\colon \Sigma_i^* \to \Sigma_{io}^* = (\cup_{j=1}^{n_i} \Sigma_{ijo})^*$ *be the projection. Assume that* $\Sigma_{ijo} = \Psi^i_{jj'}(\Sigma_{ij'o})$. *Then,* $\Psi^i_{jj'}(Q(s)) = Q(\Psi^i_{jj'}(s))$.

PROOF. We show that $\Psi^i_{jj'}(Q(s)) = Q(\Psi^i_{jj'}(s))$ by induction. Let $s = \epsilon$, then $\Psi^i_{jj'}(Q(\epsilon)) = \epsilon = Q(\Psi^i_{jj'}(\epsilon))$.

Now, we assume that $\Psi^i_{jj'}(Q(s)) = Q(\Psi^i_{jj'}(s))$ holds, and show that $\Psi^i_{jj'}(Q(sa)) = Q(\Psi^i_{jj'}(sa))$. Since $\Psi^i_{jj'}(Q(sa)) = \Psi^i_{jj'}(Q(s))\Psi^i_{jj'}(Q(a)) = Q(\Psi^i_{jj'}(s))\Psi^i_{jj'}(Q(a))$, we need to show that $\Psi^i_{jj'}(Q(a)) = Q(\Psi^i_{jj'}(a))$. We have the following cases.

Case 1: If $a \in \Sigma_i \setminus (\Sigma_{ij} \cup \Sigma_{ij'})$, then $\Psi^i_{jj'}(Q(a)) = Q(a)$ and $Q(\Psi^i_{jj'}(a)) = Q(a)$. Hence, $\Psi^i_{jj'}(Q(a)) = Q(\Psi^i_{jj'}(a))$.

Case 2: If $a \in (\Sigma_{ij} \cup \Sigma_{ij'}) \cap \Sigma_{io}$ and $a \in \Sigma_{ijo} \cap \Sigma_{gi}$, then $a \in \Sigma_{ij'o}$, and we thus have that $\Psi^i_{jj'}(Q(a)) = \Psi^i_{jj'}(a) = a$ and $Q(\Psi^i_{jj'}(a)) = Q(a) = a$. If $a \in \Sigma_{ijo} \setminus \Sigma_{gi}$, let $\Psi^i_{jj'}(a) = b$. It then follows from $\Sigma_{ijo} = \Psi^i_{jj'}(\Sigma_{ij'o})$ that $b \in \Sigma_{ij'o}$. Then $\Psi^i_{jj'}(Q(a)) = \Psi^i_{jj'}(a) = b$ and $Q(\Psi^i_{jj'}(a)) = Q(b) = b$, which proves that $\Psi^i_{jj'}(Q(a)) = Q(\Psi^i_{jj'}(a))$.

Case 3: If $a \in (\Sigma_{ij} \cup \Sigma_{ij'}) \cap (\Sigma_i \setminus \Sigma_{io})$. If $a \in (\Sigma_{ij} \setminus \Sigma_{ijo} \cap \Sigma_{gi}$, then $a \in \Sigma_{ij'} \setminus \Sigma_{ij'o}$, and we get that $\Psi^i_{jj'}(Q(a)) = \Psi^i_{jj'}(\epsilon) = \epsilon$ and $Q(\Psi^i_{jj'}(a)) = Q(a) = \epsilon$. If $a \in (\Sigma_{ij} \setminus \Sigma_{ijo}) \setminus \Sigma_{gi}$, we assume that $\Psi^i_{jj'}(a) = b$, and we have that $b \in \Sigma_{ij'} \setminus \Sigma_{ij'o}$ by the assumption that $\Sigma_{ijo} = \Psi^i_{jj'}(\Sigma_{j'o})$. Then $\Psi_{jj'}(Q(a)) = \Psi_{jj'}(\epsilon) = \epsilon$ and $Q(\Psi_{jj'}(a)) = Q(b) = \epsilon$. Hence, $\Psi^i_{jj'}(Q(a)) = Q(\Psi^i_{jj'}(a))$.

Therefore, we have $\Psi^i_{jj'}(Q(s)) = Q(\Psi^i_{jj'}(s))$. □

Finally, we need a lemma stating that if we have a set of similar plants, the underlying coordinated plant languages form a set of similar languages. We introduce notation L_{ij} for group languages $L(\mathbf{G}_{ij})$, L_k for the top coordinator language $L(\mathbf{G}_k)$, L_{k_i} for the group coordinator language $L(\mathbf{G}_{k_i})$, and finally L_{ij+k+k_i} for the coordinated systems $L_{ij} \| L_k \| L_{k_i}$ in the group $\mathcal{G}_i$.

Lemma 9 *Assume that plant languages* $L(\mathbf{G}_{ij})$ *of group* $\mathcal{G}_i$ *are similar. If the combined coordinator* $\mathbf{G}_k \| \mathbf{G}_{k_i}$ *is symmetric for group symmetry maps* $\Psi^i_{jj'}$, *then coordinated plant languages of group* $\mathcal{G}_i$ *remain similar, i.e.* $\Psi^i_{jj'}(L_{ij+k+k_i}) = L_{ij'+k+k_i}$, *where* $L_{ij+k+k_i} = L(\mathbf{G}_{ij}) \| L(\mathbf{G}_k) \| L(\mathbf{G}_{k_i})$.

PROOF. According to Lemma 4 we have $\Psi^i_{jj'}(L_{ij+k+k_i}) = \Psi^i_{jj'}(L_{ij}) \| \Psi^i_{jj'}(L_k \| L_{k_i}) = L_{ij'} \| L_k \| L_{k_i} = L_{ij'+k+k_i}$, where the symmetry of the combined coordinator, i.e. $\Psi^i_{jj'}(L_k \| L_{k_i}) = L_k \| L_{k_i}$ is used. □

We are now ready to shown that local supervisors in the same group are similar, which means that it is sufficient to apply the group symmetry map Ψ^i_{j1} to compute (j-th) local supervisor S_{ij}, $j = 1, \ldots, n_i$ of group i from the first (template) local supervisor S_{i1} of group i. Note that algorithmically we simply need to replace the local (private events) of S_{i1}, i.e. every $\sigma \in \Sigma_{pij}$, by its corresponding local event of S_{ij}, which is defined the rewriting bijection $\psi^i_{jj'}(\sigma)$.

Theorem 10 *Consider a DES similar by groups* $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_\ell\}$ *with* $\mathcal{G}_i = \{\mathbf{G}_{i1}, \mathbf{G}_{i2}, \ldots, \mathbf{G}_{in_i}\}$ *and global specification* $K \subseteq L_m(\mathbf{G})$ *that is two-level conditional decomposable with respect to alphabets* $\Sigma_1, \Sigma_2, \ldots, \Sigma_\ell$, Σ_{k_i}, *and* Σ_k, *and conditionally symmetric by group. Assume that* $\Sigma_{ijo} = \Psi^i_{jj'}(\Sigma_{ij'o})$ *and* $\Sigma_{ijc} = \Psi^i_{jj'}(\Sigma_{ij'c})$. *Then the local supervisors in the same group are similar and can be obtained by a symmetry mapping from a template* $\mathbf{S}_{i1}$ *for* $i \in [1, \ell]$, *i.e.,*

$$(\forall j \in [1, n_i])\; \mathbf{S}_{ij} = \Psi^i_{j1}(\mathbf{S}_{i1}), \qquad (11)$$

where $L_m(\mathbf{S}_{i1}) = \sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})$.

PROOF. Recall from (7) the expression for S_{ij}. We first show $\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})) \subseteq P_{ij+k+k_i}(K) \cap L_{ij+k+k_i}$ and then that $\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))$ is controllable and relatively observable with respect to L_{ij+k+k_i}.

Let $s \in \Psi^i_{jj'}(\mathbf{S}_{i1})$. By $\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})) \subseteq \Psi^i_{j1}(P_{i1+k+k_i}(K) \cap L_{i1+k+k_i})$ we have $s \in \Psi^i_{j1}(P_{i1+k+k_i}(K) \cap L_{i1+k+k_i})$. Since K is symmetric by group, we have that $s \in \Psi^i_{j1}(P_{i1+k+k_i}(K)) = P_{ij+k+k_i}(K)$ by definition. By the group similarity of local plants and symmetry of the coordinators, Lemma 9 gives that $s \in \Psi^i_{j1}(L_{i1+k+k_i}) = L_{ij+k+k_i}$. Thus we have $s \in P_{ij+k+k_i}(K) \cap L_{ij+k+k_i}$.

Next we prove that $\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))$ is controllable with respect to L_{ij+k+k_i}.

Let $s \in \overline{\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))}$ and $s\sigma \in L_{ij+k+k_i}$ for $\sigma \in \Sigma_u$. Then

$$
\begin{aligned}
& s \in \Psi^i_{j1}\overline{(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))} \\
& \Rightarrow \Psi^i_{1j}(s) \in \Psi^i_{1j}\Psi^i_{j1}\overline{(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))} \\
& \Rightarrow \Psi^i_{1j}(s) \in \overline{(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))}
\end{aligned}
$$

From $s\sigma \in L_{ij+k+k_i}$ we obtain $\Psi^i_{1j}(s\sigma) \in \Psi^i_{1j}(L_{ij+k+k_i}) = L_{i1+k+k_i}$. Since $\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})$ is controllable with respect to L_{i1+k+k_i} and $\Psi^i_{1j}(\sigma)$ is uncontrollable due to the assumption that controllability status is preserved by symmetry map, we have that $\Psi^i_{1j}(s)\Psi^i_{1j}(\sigma) = \Psi^i_{1j}(s\sigma) \in \overline{\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})}$. Thus

$$
\begin{aligned}
& \Psi^i_{j1}(\Psi^i_{1j}(s\sigma)) = s\sigma \in \Psi^i_{j1}\overline{(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))} \\
& = \overline{\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))}.
\end{aligned}
$$

Now we show that $\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))$ is relatively observable with respect to L_{ij+k+k_i} and Q. Let $sa \in \overline{\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))}$ and $s' \in \overline{P_{ij+k+k_i}(K)}$ such that $Q(s) = Q(s')$ and $s'a \in L_{ij+k+k_i}$. We show that $s'a \in \overline{\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))}$. Then, there exist a string $\tilde{s} \in \Sigma^*$ and a string $\tilde{a} \in \Sigma$ such that $\tilde{s}\tilde{a} \in \overline{\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})}$ and $\Psi^i_{j1}(\tilde{s}\tilde{a}) = sa$. Since K is symmetric by group, we have $P_{ij+k+k_i}(K) = \Psi^i_{j1}(P_{i1+k+k_i}(K))$. Since $s' \in P_{ij+k+k_i}(K)$, there exists a string $\tilde{s'} \in P_{i1+k+k_i}(K)$ with $\Psi^i_{j1}(\tilde{s'}) = s'$. The agents in the same group are similar and the combined coordinator is symmetric, so we obtain by Lemma 9 that $\Psi^i_{j1}(s'a) = \tilde{s'}\tilde{a} \in \Psi^i_{j1}(L_{ij+k+k_i}) = L_{i1+k+k_i}$. By Lemma 8 and $Q(s) = Q(s')$, it follows that $\Psi^i_{j1}(Q(s)) = Q(\Psi^i_{j1}(s)) = Q(\tilde{s}) = Q(\tilde{s'}) = Q(\Psi^i_{j1}(s')) = \Psi^i_{j1}(Q(s'))$. It follows from relative observability of $\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})$ with respect to L_{i1+k+k_i} and Q that $\tilde{s'}\tilde{a} \in \overline{\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i})}$. Therefore, $\Psi^i_{j1}(\tilde{s'}\tilde{a}) = s'a \in \overline{\Psi^i_{j1}(\sup CO(P_{i1+k+k_i}(K), L_{i1+k+k_i}))}$. Thus, $\mathbf{S}_{ij} \subseteq \Psi^i_{j1}(\mathbf{S}_{i1})$ is now proved. Recall that due to (2) this completes the proof. □

Notice that the computations of local supervisors involved in the above theorem are independent of the number n_i. By (11), computing $\mathbf{S}_{ij}$ for all $j \in [1, n_i]$ requires the computing of $\mathbf{S}_{i1}$, which requires the computing of $P_{i1+k+k_i(K)}$ and L_{i1+k+k_i}, both of which are independent of n_i. Therefore, the computation of the local supervisors $\mathbf{S}_{i1}, \ldots, \mathbf{S}_{in_i}$ is independent of the number n_i of agents, and thus solves the state explosion problem of the supervisory control theory. Moreover, the template $\mathbf{S}_{i1}$ in Theorem 10 can be replaced by an arbitrary local supervisor $\mathbf{S}_{ij}$ in group i. Namely, we can compute an arbitrary local supervisor in one group and take this local supervisor as a template.

Remark 11 *Theorem 10 requires that the global specification K should be two-level conditional decomposable and symmetric by group. Two-level conditional decomposability for K is not restrictive since there always exist alphabets $\Sigma_{k_i} \subseteq \Sigma_i$ and $\Sigma_k \subseteq \Sigma$ such that the global specification is two-level conditionally decomposable. For the symmetry by group condition, if K is not symmetric by group, according to Proposition 2 the set of similar sublanguages for each group can be computed in a modular way under a given condition. Therefore, given a similar discrete-event system and a specification that is symmetric by group, we can get that the local supervisors in the same group are similar.*

4 Illustrative example

Consider a small factory system, adapted from Wonham and Cai (2019). As displayed in Fig. 2 small factory consists of three input machines $\mathbf{G_1}$, $\mathbf{G_2}$, $\mathbf{G_3}$ and two output machines $\mathbf{G_4}$ and $\mathbf{G_5}$ linked by a buffer with capacities three in the middle. The generators are shown in Fig. 2. Let $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \{l1, l3, 3\} \dot\cup \{l2\}$ for $l \in [1,5]$, $\Sigma = \Sigma_o \dot\cup \Sigma_{uo} = \{l1, l3, 3\} \dot\cup \{l2\}$, and $\Sigma_g = \{3\}$. The alphabet of agents in the input group is $\Sigma_1 = \cup_{j=1}^3 \Sigma_{1j} = \cup_{j=1}^3 \{j1, j2, j3, 3\}$ and the alphabets of agents in the output group are $\Sigma_2 = \cup_{j=4}^5 \Sigma_{2j} = \cup_{j=4}^5 \{j1, j2, j3, 3\}$. Based on their different roles, the machines are divided into two groups: $\mathcal{G}_1 = \{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G_3}\}$ and $\mathcal{G}_2 = \{\mathbf{G}_4, \mathbf{G}_5\}$. For different groups, the group symmetry map is given below.

$\Psi^1_{jj'}(j0) = j'0; \Psi^1_{jj'}(j1) = j'1; \Psi^1_{jj'}(j2) = j'2; j, j' = 1, 2, 3;$

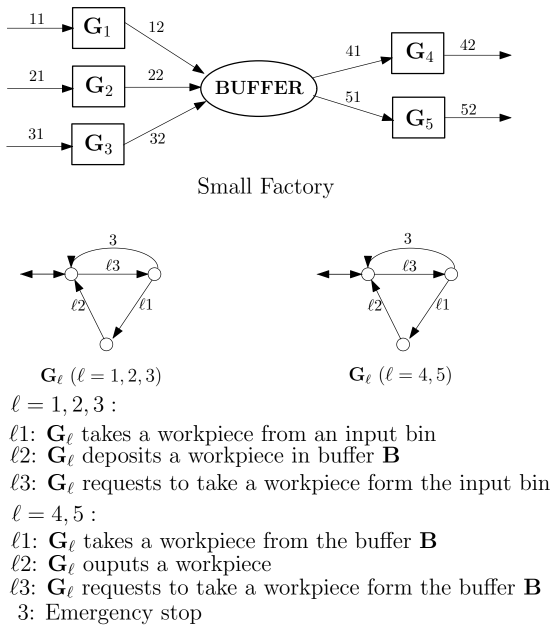

Figure 2. System configuration and plant generator models

$\Psi^2_{jj'}(j0) = j'0; \Psi^2_{jj'}(j1) = j'1; \Psi^2_{jj'}(j2) = j'2; j, j' = 4,5;$
$\Psi^1_{jj'}(3) = \Psi^2_{jj'}(3) = 3;$

The control specifications are as follows:
1. Agents in the input group to input a workpiece have priority over agents in the output group to take a workpiece if an emergency stop happens.
2. The agent in the output group that first requests to take a workpiece is first granted the workpiece.

These specifications $\mathbf{K_1}$ and $\mathbf{K_2}$ are in Fig.3. The meanings of each state of $\mathbf{K}_1$ are: state x means all agents are in normal state; state y denotes agents in the input group have priority to take a workpiece (by events 11, 21, 31) after emergency stop (event 3) happens. The meanings of each state of $\mathbf{K}_2$ are: state 1 means two agents in the output group are allowed to take the workpiece; state 2 means agent $\mathbf{G_4}$ or $\mathbf{G_5}$ request to take a workpiece (event 43 or 53 happens); state 3 means agent $\mathbf{G_4}$ requests to take a workpiece from the buffer (event 43) since $\mathbf{G_5}$ has requested to take a workpiece at state 2, which leads to that $\mathbf{G_5}$ is first granted the workpiece (event 51 is enabled at state 3); similarly, at state 4, $\mathbf{G_4}$ is first granted the workpiece (event 41 is enabled) since $\mathbf{G_4}$ first requests to take a workpiece (event 43 happens at state 2).

The global specification is $\mathbf{K} = \mathbf{K_1} \| \mathbf{K_2}$. Let $\Sigma_k = \{3, 41, 51\}$. We have $K = K_1 \| K_2 = P_{1+k}(K) \| P_{2+k}(K)$, where $P_{i+k} : \Sigma^* \to (\Sigma_i \cup \Sigma_k)^*$ for $i = 1, 2$. The high-level coordinator $\mathbf{G}_k$ is $\mathbf{G}_k = P_k(\mathbf{G}_1) \| P_k(\mathbf{G}_2)$, cf. Fig.4. Then we need to compute the low-level coordinators for each group. For the input group, let $\Sigma_{k_1} = \{3, 11, 21, 31\}$. We thus get that $K_1 = P_{11+k+k_1}(K_1) \| P_{12+k+k_1}(K_1) \| P_{13+k+k_1}(K_1)$, where $P_{1j+k+k_1}(K_1)$ are the local specifications for the input group with $P_{1j+k+k_1} : \Sigma^* \to (\Sigma_{1j} \cup \Sigma_k \cup \Sigma_{k_1})^*$ and $j = 1, 2, 3$, cf. Fig. 4. The corresponding low-

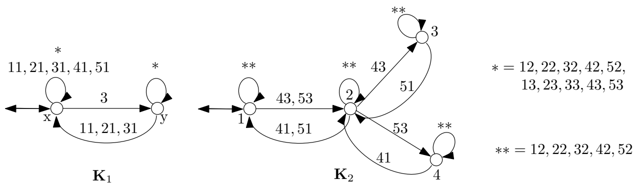

Figure 3. Specification generator models

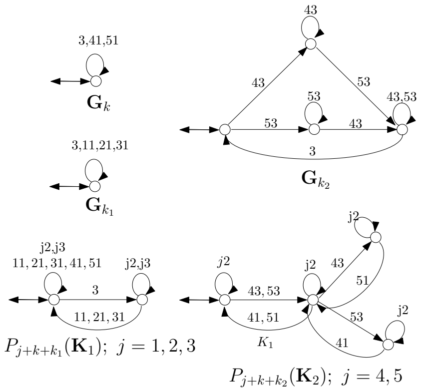

Figure 4. Coordinators and local specification generator models

level coordinator $\mathbf{G}_{k_1}$ for the input group is $\mathbf{G}_{k_1} = P_{k_1}(\mathbf{G}_1) \| P_{k_1}(\mathbf{G}_2) \| P_{k_1}(\mathbf{G}_3)$, cf. Fig.4. It is symmetric. For the output group, we choose $\Sigma_{k_2} = \{3, 43, 53\}$ and have that $K_2 = P_{21+k+k_2}(K_2) \| P_{22+k+k_2}(K_2)$, where $P_{2j+k+k_2}(K_2)$ are the local specifications, where $P_{2j+k+k_1} : \Sigma^* \to (\Sigma_{2j} \cup \Sigma_k \cup \Sigma_{k_2})^*$ and $j = 4, 5$, cf. Fig. 4. The coordinator $\mathbf{G}_{k_2}$ for the output group is $\mathbf{G}_{k_2} = P_{k_2}(\mathbf{G}_4) \| P_{k_2}(\mathbf{G}_5)$, cf. Fig.4. It is also symmetric. Since $\Psi^1_{23}(P_{11+k+k_1}(K_1)) = P_{11+k+k_1}(K_1)$, $\Psi^1_{13}(P_{12+k+k_1}(K_1)) = P_{12+k+k_1}(K_1)$, and $\Psi^1_{12}(P_{13+k+k_1}(K_1)) = P_{13+k+k_1}(K_1)$, the input group specification $P_{11+k+k_1}(K_1)$, $P_{12+k+k_1}(K_1)$, and $P_{13+k+k_1}(K_1)$ are similar. Since $\Psi^2_{45}(P_{24+k+k_2}(K_2)) = P_{25+k+k_2}(K_2)$, $P_{24+k+k_2}(K_2)$ and $P_{25+k+k_2}(K_2)$ are similar. Thus, K is symmetric by group. By Theorem 10 local supervisors in each group are similar, i.e. it suffices to compute one per group, cf. Fig. 5..

5 Conclusion

An efficient modular approach to compute a set of template supervisors for plants composed of systems similar by group is proposed based on local computations of supremal symmetric sublanguages and on decomposing specifications. It is shown that local supervisors for the components of a group are similar, and can be computed by a symmetry map from a template supervisor. In a future we plan to extend our approach to real-time systems.

References

Agarwal, M., Biswas, S., Nandi, S., 2019. Discrete event system framework for fault diagnosis with measurement in-

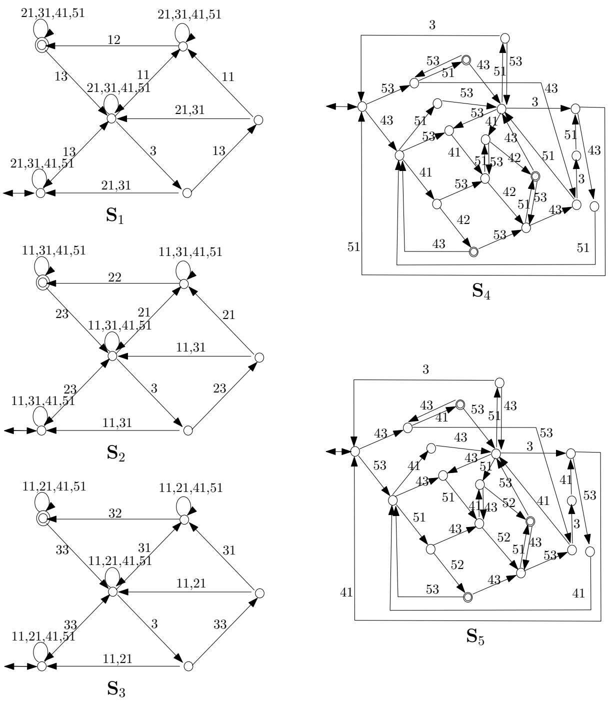

Figure 5. Local supervisors

consistency: Case study of rogue dhcp attack. IEEE/CAA J. Autom. Sin. 6, 789–806.

Amini, A., Asif, A., Mohammadi, A., 2020. Formation-containment control using dynamic event-triggering mechanism for multi-agent systems. IEEE/CAA J. Autom. Sin. 7, 1235–1248.

Basu, S., Kumar, R., 2021. Control of non-deterministic systems with mu-calculus specifications using quotienting. IEEE/CAA Journal of Automatica Sinica 8, 953–970.

Cai, K., Zhang, R., Wonham, W.M., 2015. Relative observability of discrete-event systems and its supremal sublanguages. IEEE Trans. Automat. Contr. 60, 659–670.

Cassandras, C., Lafortune, S., 2008. Introduction to discrete event systems. Springer.

Ding, Z., Qiu, H., Yang, R., Jiang, C., Zhou, M., 2019. Interactive-control-model for human-computer interactive system based on petri nets. IEEE Trans. Autom. Sci. Eng. 16, 1800–1813.

Du, N., Hu, H., Zhou, M.C., 2020. Robust deadlock avoidance and control of automated manufacturing systems with assembly operations using petri nets. IEEE Trans. Autom. Sci. Eng. 17, 1961–1975.

Eyzell, J.M., Cury, J.E.R., 2001. Exploiting symmetry in the synthesis of supervisors for discrete event systems. IEEE Trans. Automat. Contr. 46, 1500–1505.

Jiao, T., Gan, Y., Xiao, G., Wonham, W.M., 2017. Exploiting symmetry of state tree structures for discrete-event systems with parallel components. Int. J. Control 90, 1639–1651.

Komenda, J., Masopust, T., van Schuppen, J., 2012. On conditional decomposability. Syst. Control. Lett. 61, 1260–1268.

Komenda, J., Masopust, T., van Schuppen, J.H., 2013. Multilevel coordination control of modular discrete-event systems, in: IEEE CDC, pp. 6323–6328.

Liu, Y., Cai, K., Li, Z., 2019. On scalable supervisory control of multi-agent discrete event systems. Automatica 108, 108460.

Luo, J., Ni, H., Zhou, M., 2015. Control program design for automated guided vehicle systems via petri nets. IEEE Trans. Syst. Man Cybern.: Syst. 45, 44–55.

Rohloff, K., Lafortune, S., 2006. The verification and control of interacting similar discrete-event systems. SIAM J. Control Optim. 45, 634–667.

Su, R., Lennartsson, B., 2017. Control protocol synthesis for multi-agent systems with similar actions instantiated from agent and requirement templates. Automatica 79, 244–255.

Wang, W., Su, R., Lin, L., Gong, C., 2019. Model checking in isomorphic module systems. IEEE Trans. Automat. Contr. 64, 728–735. doi:10.1109/TAC.2018.2844683.

Willner, Y., Heymann, M., 1991. Supervisory control of concurrent discrete-event systems. Int. J. Control 54, 1143–1169.

Wonham, W.M., Cai, K., 2019. Supervisory control of discrete-event systems. Springer.

Wu, N., Zhou, M., 2007a. Deadlock resolution in automated manufacturing systems with robots. IEEE Trans. Autom. Sci. Eng. 4, 474–480.

Wu, N.Q., Zhou, M.C., 2007b. Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking. IEEE ASME Trans Mechatron 12, 63–72.

Wurman, P., D'Andrea, R., Mountz, M., 2008. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Magazine 29, 9 – 19.