Automatica 46 (2010) 1489–1494

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Abstraction-based verification of codiagnosability for discrete event systems[☆]

K. Schmidt*

Department of Electronic and Communication Engineering, Cankaya University, 06530 Ankara, Turkey

ARTICLE INFO

Article history:
Received 22 January 2010
Received in revised form
7 April 2010
Accepted 13 May 2010
Available online 1 July 2010

Keywords:
Discrete event systems
Failure diagnosis
Codiagnosability
Abstraction
Verification

ABSTRACT

In this paper, we investigate the verification of *codiagnosability* for discrete event systems (DES). That is, it is desired to ascertain if the occurrence of system faults can be detected based on the information of multiple local sites that partially observe the overall DES. As an improvement of existing codiagnosability tests that resort to the original DES with a potentially computationally infeasible state space, we propose a method that employs an *abstracted* system model on a smaller state space for the codiagnosability verification. Furthermore, we show that this abstraction can be computed without explicitly evaluating the state space of the original model in the practical case where the DES is composed of multiple subsystems.

1. Introduction

The failure diagnosis for discrete event systems (DES) is concerned with the detection of undesired (faulty) system behavior based on the *partial observation* of the system evolution by one or multiple *diagnosers*. In a *centralized* setting, the failure diagnosis is carried out by a single diagnoser (Hashtrudi Zad, Kwong, & Wonham, 2003; Paoli & Lafortune, 2005; Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995; Yoo & Garcia, 2008), while multiple diagnosers operate at several local sites without exchanging information in *decentralized* methods such as Debouk and Teneketzis (2000), Qiu and Kumar (2006), Su and Wonham (2005) and Wang, Yoo, and Lafortune (2007). *Distributed* diagnosis approaches (Qiu & Kumar, 2008; Ricker & van Schuppen, 2001) add communication to the decentralized setting. Furthermore, the practical case of DES that are *composed* of multiple subsystems is considered in Debouk, Malik, and Brandin (2002), García, Correcher, Morant, Quiles, and Blasco (2005), Takai (2008) and Zhou, Kumar, and Sreenivas (2008), whereby the additional system structure allows for more efficient computations. In this paper, we investigate the decentralized setting for composed DES.

Generally, the diagnosis problem can be solved if each fault can be uniquely detected after the occurrence of a finite number of events. In the decentralized setting, this property is captured by the notion of *codiagnosability*. If a DES that is observed by *diagnosers* at multiple local sites is codiagnosable, then the occurrence of any fault can be identified by at least one diagnoser solely based on its local observation (Debouk et al., 2002; Debouk & Teneketzis, 2000; Qiu & Kumar, 2006; Wang et al., 2007; Zhou et al., 2008). This setting is particularly useful for DES, where the diagnostic decision has to be made by the local site that collects the respective information (Qiu & Kumar, 2006).

In Debouk et al. (2002), Debouk and Teneketzis (2000) and Wang et al. (2007), codiagnosability is investigated based on the characterization of faulty behavior in terms of *failure events*, and polynomial time algorithms to verify event-based codiagnosability are provided in Qiu and Kumar (2004) and Wang et al. (2007). In contrast, faulty system behavior is captured by a *language specification* in Qiu and Kumar (2006) and Zhou et al. (2008). Polynomial time algorithms for the verification of language-based codiagnosability are elaborated in Qiu and Kumar (2006) and applied to DES that are composed of multiple subsystems in Zhou et al. (2008). However, in all cases, the codiagnosability verification requires the explicit computation of the overall system, which makes it infeasible to apply the existing methods to large-scale DES.

In this paper, we adapt the idea of *abstraction-based language-diagnosability* introduced in our previous work (Schmidt, 2010) to the codiagnosability verification in the framework of Qiu and Kumar (2006) and Zhou et al. (2008). That is, we compute an abstracted system model on a smaller state space than the original model using the natural projection on a subset of the system alphabet. As is discussed in Section 3, this abstraction allows the codiagnosability verification with an efficiently reduced

[☆] This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor George Yin under the direction of Editor Ian R. Petersen.

[*] Tel.: +90 312 2844500/4136; fax: +90 312 2848043.
E-mail address: schmidt@cankaya.edu.tr.

0005-1098/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2010.06.010

computational effort if certain sufficient conditions for the natural projection hold. Moreover, we show in Section 4 how the abstracted model can be computed without enumerating the overall system state space if the DES is composed of multiple components. A manufacturing unit demonstrates the benefits of our method.

2. Preliminaries

2.1. Basic notation

We denote the set of all finite strings over a finite alphabet Σ including the empty string ϵ as Σ^*, characterize the length of a string $s \in \Sigma^*$ by $|s|$ and write $s_1 \leq s$ for $s, s_1 \in \Sigma^*$ if s_1 is a prefix of s. A subset $L \subseteq \Sigma^*$ is denoted as a *language*. L is *prefix-closed* if $L = \overline{L} := \{s_1 \in \Sigma^* | \exists s \in L \text{ s.t. } s_1 \leq s\}$.

The *natural projection* $p : \Sigma^* \to \hat{\Sigma}^*$, $\hat{\Sigma} \subseteq \Sigma$ is defined iteratively: (1) let $p(\epsilon) := \epsilon$; (2) for $s \in \Sigma^*, \sigma \in \Sigma$, let $p(s\sigma) := p(s)\sigma$ if $\sigma \in \hat{\Sigma}$, or $p(s\sigma) := p(s)$ otherwise. The associated inverse projection is $p^{-1} : \hat{\Sigma}^* \to 2^{\Sigma^*}, p^{-1}(t) := \{s \in \Sigma^* | p(s) = t\}$. A useful property of projections that was introduced in the context of hierarchical supervisory control is the *observer condition* (Wong & Wonham, 1996).

Definition 1. Let $L = \overline{L} \subseteq \Sigma^*$ be a prefix-closed language. The projection $p : \Sigma^* \to \hat{\Sigma}^*$ is an observer if

$$\begin{aligned}&(\forall s \in L) \quad (\forall t \in \hat{\Sigma}^* \text{ s.t. } p(s)t \in p(L))\\ &\Rightarrow \exists u \in \Sigma^* \text{ s.t. } su \in L \quad \text{and} \quad p(su) = p(s)t.\end{aligned} \tag{1}$$

In this paper, a DES is modeled by a *finite automaton* $G = (X, \Sigma, \delta, x_0)$ with the *states* X, the *alphabet* Σ, the partial *transition function* $\delta : X \times \Sigma \to X$ and the *initial state* x_0. The *closed language* $L(G)$ of G and the *synchronous composition* $G_1 \parallel G_2$ of two automata G_1 and G_2 are defined in the usual way (Cassandras & Lafortune, 2006).

2.2. Codiagnosability

Adopting the framework in Qiu and Kumar (2006) and Zhou et al. (2008), we represent a partially observed DES by an automaton $G = (X, \Sigma, \delta, x_0)$, whose behavior is seen through local *observation masks* $M_i : \Sigma \to \Delta_i \cup \{\epsilon\}$ at multiple local sites $i \in \mathcal{I} = \{1, \ldots, m\}$. In this context, each local mask $M_i, i \in \mathcal{I}$ maps events $\sigma \in \Sigma$ to their local observation $M_i(\sigma) \in \Delta_i \cup \{\epsilon\}$, where Δ_i is the *set of observations* at site i. For each $i \in \mathcal{I}$, we denote $\Sigma_{i,o} := \{\sigma \in \Sigma | M_i(\sigma) \neq \epsilon\}$ as the set of *observable* events. Furthermore, M_i can be recursively extended to strings by defining $M_i(s\sigma) = M_i(s)M_i(\sigma)$ for $s \in \Sigma^*$ and $\sigma \in \Sigma$. The corresponding inverse map $M_i^{-1} : \Delta_i^* \to 2^{\Sigma^*}$ is defined such that for $d \in \Delta_i^*, M_i^{-1}(d) = \{s \in \Sigma^* | M_i(s) = d\}$.

Analogous to Qiu and Kumar (2006), Yoo and Garcia (2008) and Zhou et al. (2008), we represent a fault by the violation of a given prefix-closed specification language $K = \overline{K} \subseteq L(G)$. That is, we employ the natural assumption that a string $s \in L(G)$ (and any prefix of s) is regarded as correct as long as $s \in K = \overline{K}$, while s is decided to be faulty as soon as $s \notin K$. Since our goal is to perform decentralized diagnosis without communication, it is desired that each deviation from the correct behavior K can be uniquely inferred from the observations of at least one local site $i \in \mathcal{I}$ through its mask M_i. The following definition of *codiagnosability* as employed in Qiu and Kumar (2006) and Zhou et al. (2008) formally states this objective.

Definition 2. Let G be a DES over the alphabet Σ, let $K = \overline{K} \subseteq L(G)$ be a prefix-closed specification language and assume m local sites with their observation masks $M_i, i \in \mathcal{I}$. K is codiagnosable for G and $M_i, i \in \mathcal{I}$ if

$$\begin{aligned}&(\exists n \in \mathbb{N}) \quad (\forall s \in L(G) - K)\\ &(\forall st \in L(G) \text{ s.t. } |t| \geq n \text{ or } st \text{ deadlocks})\\ &\Rightarrow (\exists i \in \mathcal{I}) \quad (\forall u_i \in M_i^{-1}M_i(st) \cap L(G), u_i \notin K).\end{aligned} \tag{2}$$

The smallest n that satisfies (2) is denoted as the *worst-case detection delay*.

Definition 2 implies that every faulty string in $L(G) - K$ can be uniquely distinguished from the correct strings in K by at least one local site after a finite *detection delay*, i.e., the occurrence of a bounded number of events. Defining p_G as the number of states of G, p_C as the number of states of a specification automaton C with $L(C) = K$, and q_Σ as the number of events in Σ, codiagnosability can be verified with a computational complexity of $\mathcal{O}(p_G \cdot p_C^{m+1} \cdot q_\Sigma^{m+1})$ (Qiu & Kumar, 2006).

3. Abstraction-based codiagnosability test

The computational complexity of the codiagnosability verification strongly depends on the state counts of the automata G and C. Since both automata have to be enumerated explicitly, the method in Qiu and Kumar (2006) is not applicable to large-scale DES. In this paper, we adopt ideas from the abstraction-based supervisory control (Schmidt, Moor, & Perk, 2008) to develop the abstraction-based codiagnosability verification.

3.1. Problem statement

We assume that the model G and the masks $M_i, i \in \mathcal{I}$ are given as described in Section 2.2. However, we are interested in the practical case, where a reduced specification $K' \subseteq \Sigma'^*$ with $\Sigma' \subseteq \Sigma$ is given[1] such that the overall specification $K \subseteq \Sigma^*$ evaluates as

$$K = K' \parallel L(G) \subseteq L(G). \tag{3}$$

We then propose to exploit this property of K to perform the codiagnosability verification based on an *abstracted model* $\hat{G}$ and an *abstracted specification* $\hat{K} \subseteq L(\hat{G})$ over an *abstraction alphabet* $\hat{\Sigma} \subseteq \Sigma$. In order to capture the relevant behavior specified by K', we require that $\Sigma' \subseteq \hat{\Sigma}$.[2] Then, $\hat{G}$ and $\hat{K}$ are determined using the natural projection $p : \Sigma^* \to \hat{\Sigma}^*$ such that

$$L(\hat{G}) := p(L(G)) \quad \text{and} \quad \hat{K} := K' \parallel L(\hat{G}) = p(K). \tag{4}$$

We define the *abstracted observation masks* $\hat{M}_i : \hat{\Sigma} \to \hat{\Delta}_i \cup \{\epsilon\}, i \in \mathcal{I}$, where $\hat{\Delta}_i = \{M_i(\sigma) | \sigma \in \hat{\Sigma}\}$ contains all possible observations of events in $\hat{\Sigma}$ such that, for all $\sigma \in \hat{\Sigma}, \hat{M}_i(\sigma) = M_i(\sigma)$. Using the abstracted entities $\hat{G}, \hat{K}, \hat{M}_i, i \in \mathcal{I}$, we study the following problem.

Problem 1. Let G be a model automaton, $K' \subseteq \Sigma'^*$ be a reduced specification and $M_i : \Sigma \to \Delta_i \cup \{\epsilon\}$ be local observation masks for $i \in \mathcal{I}$. Defining $\hat{G}, \hat{K}$ and $\hat{M}_i$ as above for the abstraction alphabet

[1] For example, Section 4.2 employs such specification.

[2] Additional information on the choice of $\hat{\Sigma}$ is provided in Remark 1.

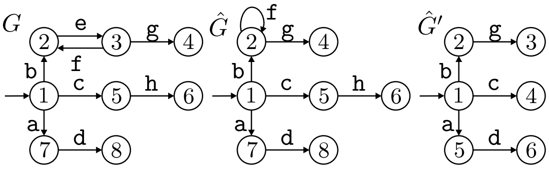

Fig. 1. Illustration of the sufficient conditions.

$\hat{\Sigma}$ with $\Sigma' \subseteq \hat{\Sigma} \subseteq \Sigma$, we want to find sufficient conditions such that

(1) codiagnosability of $\hat{K} = K' \parallel L(\hat{G})$ for $\hat{G}$ and $\hat{M}_i, i \in \mathcal{I}$ implies codiagnosability of $K := K' \parallel L(G)$ for G and $M_i, i \in \mathcal{I}$, while the abstracted model $\hat{G}$ has a smaller state space than the model G.
(2) Codiagnosability of K for G and $M_i, i \in \mathcal{I}$ also implies codiagnosability of $\hat{K}$ for $\hat{G}$ and $\hat{M}_i, i \in \mathcal{I}$.

In view of item (1), the codiagnosability verification developed in Zhou et al. (2008) can now be applied to the abstracted system with $\hat{G}$, $\hat{K}$ and $\hat{M}_i, i \in \mathcal{I}$. Furthermore, the state count $p_{\hat{G}}$ of $\hat{G}$ is smaller than p_G,[3] and the choice of $\hat{\Sigma}$ and the composition of $\hat{K}$ suggest that the number $q_{\hat{\Sigma}}$ of events in $\hat{\Sigma}$ and the state count $p_{\hat{C}}$ of $\hat{C}$ are smaller than q_Σ and p_C, respectively. Together, it is expected that the computational complexity $\mathcal{O}(p_{\hat{G}} \cdot p_{\hat{C}}^{m+1} \cdot q_{\hat{\Sigma}}^{m+1})$ for the abstraction-based codiagnosability verification is efficiently reduced compared to the original test as addressed in Section 2.2. In turn, item (2) is beneficial if the codiagnosability verification of the abstracted system fails. Then, it can be concluded that the original system is also not codiagnosable.

3.2. Sufficient conditions

In our previous work (Schmidt, 2010), *loop-preserving observers* and *consistent observation masks* were employed for the abstraction-based language-diagnosability verification. In this section, we first introduce both conditions and then show that these conditions are indeed sufficient for the solution of Problem 1(1). Finally, we provide an additional property that addresses Problem 1(2).

A loop-preserving observer is an observer as in Definition 1 such that, additionally, any arbitrarily long strings in L also appear as such strings in the abstraction $p(L)$.

Definition 3. p in Definition 1 is a loop-preserving observer (LPO) for L with bound N if for all u in (1), $|u| < N \cdot (|t| + 1)$.

As a first example, we study the model G over $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}, \mathtt{h}\}$ in Fig. 1 with the abstraction alphabet $\hat{\Sigma} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{f}, \mathtt{g}, \mathtt{h}\}$ (projection $p : \Sigma^* \to \hat{\Sigma}^*$). The resulting abstracted model is $\hat{G}$. It can be verified that the observer condition in Definition 1 is fulfilled. Furthermore, the only loop in G with the events $\mathtt{e}$ and $\mathtt{f}$ appears as a loop with $\mathtt{f}$ in $\hat{G}$. Hence, p is a loop-preserving observer. In contrast, the projection $p' : \Sigma^* \to \hat{\Sigma}'$ with $\hat{\Sigma}' = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{g}\}$ violates the LPO condition since the loop with the events $\mathtt{e}$ and $\mathtt{f}$ in G does not appear as a loop in the abstracted model $\hat{G}'$ in Fig. 1.

An observation mask M_i is consistent with the projection p if all events with the same non-empty observation are either retained or removed by the abstraction (Schmidt, 2010).

Definition 4. The observation mask $M_i : \Sigma \to \Delta_i \cup \{\epsilon\}$ with the observable events $\Sigma_{i,\mathrm{o}}$ is consistent with the natural projection $p : \Sigma^* \to \hat{\Sigma}^*$ if for any $\sigma, \sigma' \in \Sigma_{i,\mathrm{o}}$ with $M_i(\sigma) = M_i(\sigma')$, it holds that $\sigma \in \hat{\Sigma}_i \Leftrightarrow \sigma' \in \hat{\Sigma}_i$.

We consider the projection p from the previous example and two observation masks $M_i : \Sigma \to \Delta_i \cup \{\epsilon\}, i = 1, 2$. The observations are $\Delta_1 = \{\mathtt{G}\}$ with $M_1(\mathtt{g}) = \mathtt{G}$ and $M_1(\sigma) = \epsilon$ for $\sigma \in \Sigma - \{\mathtt{g}\}$ as well as $\Delta_2 = \{\mathtt{D}\}$ with $M_2(\mathtt{d}) = M_2(\mathtt{h}) = \mathtt{D}$ and $M_2(\sigma) = \epsilon$ for $\sigma \in \Sigma - \{\mathtt{d}, \mathtt{h}\}$. Then, both M_1 and M_2 are consistent with p. In particular, considering the mask M_2, the two events $\mathtt{d}$ and $\mathtt{h}$ with the same observation $\mathtt{D}$ are both included in $\hat{\Sigma}$. In contrast, M_2 is not consistent with p' from the previous example since $M_2(\mathtt{d}) = M_2(\mathtt{g})$, whereas $\mathtt{d} \in \hat{\Sigma}'$ and $\mathtt{g} \notin \hat{\Sigma}'$. That is, although both events cannot be distinguished by observation, the abstraction wrongly suggests that $\mathtt{d}$ and $\mathtt{g}$ can be told apart.

The main theorem of this section is based on Definitions 3 and 4.

Theorem 1. *Problem 1(1) is solved if p is an LPO and M_i is consistent with p for all $i \in \mathcal{I}$. Furthermore, Problem 1(2) is solved if $\Sigma_\mathrm{o} := \bigcup_{i \in \mathcal{I}} \Sigma_{i,\mathrm{o}} \subseteq \hat{\Sigma}$, i.e., all observable events are retained in the abstraction.*

That is, if the conditions in Definitions 3 and 4 are fulfilled, then codiagnosability of the original system can be verified by means of the codiagnosability test for the abstracted system. Moreover, it is sufficient to preserve all available observations in the abstraction such that the reverse statement holds.

We now apply Theorem 1 to G and $\hat{G}$ in Fig. 1 using the masks M_1 and M_2 defined above. That is, both the LPO condition for p and consistency of M_1 and M_2 hold. Furthermore, we introduce the specification $K' = \{\ \}$ over the alphabet $\Sigma' = \{\mathtt{b}, \mathtt{c}\} \subseteq \hat{\Sigma}$ such that any strings with the events $\mathtt{b}$ or $\mathtt{c}$ are faulty. Verifying codiagnosability for the abstraction, it turns out that the faulty string $\mathtt{b}$ can be uniquely identified via the abstracted mask $\hat{M}_1$ after $\mathtt{g}$ occurred, while the faulty string $\mathtt{c}$ cannot be distinguished from correct strings by any mask. Hence, the violation of codiagnosability for the abstraction implies that the original system is not codiagnosable according to Theorem 1 ($\Sigma_\mathrm{o} = \{\mathtt{d}, \mathtt{g}, \mathtt{h}\} \subseteq \hat{\Sigma}$). Finally, we modify M_2 such that $\Delta_2 = \{\mathtt{D}, \mathtt{H}\}$ and $M_2(\mathtt{d}) = \mathtt{D}, M_2(\mathtt{h}) = \mathtt{H}$. Then, the failure string $\mathtt{c}$ is detected via $\hat{M}_2$ as soon as $\mathtt{h}$ occurred in the abstraction. In this case, codiagnosability for the abstraction implies codiagnosability for the original system since all conditions in Theorem 1 hold.

The proof of Theorem 1 relies on the following lemmas.

Lemma 1. *Let p, M_i and $\hat{M}_i$ be as defined above and let $\hat{p}_i : \Delta_i^* \to \hat{\Delta}_i^*$ be the natural projection. If M_i is consistent with p, then $\hat{p}_i M_i(s) = \hat{M}_i p(s)$ for all $s \in \Sigma^*$.*

Proof. Let $\hat{\Sigma}_{i,\mathrm{o}} := \Sigma_{i,\mathrm{o}} \cap \hat{\Sigma}$ and $s = u_1\sigma_1 \cdots \sigma_p u_{p+1}$ with $\sigma_j \in \hat{\Sigma}_{i,\mathrm{o}}, j = 1, \ldots, p$, and $u_j \in (\Sigma - \hat{\Sigma}_{i,\mathrm{o}})^*, j = 1, \ldots, p+1$. Since M_i is consistent with $p, i \in \mathcal{I}$, it holds that $\hat{p}_i M_i(s) = \hat{p}_i\left(M_i(u_1)M_i(\sigma_1) \cdots M_i(\sigma_p)M_i(u_{p+1})\right) = M_i(\sigma_1) \cdots M_i(\sigma_p) = \hat{M}_i(\sigma_1) \cdots \hat{M}_i(\sigma_p) = \hat{M}_i\left(p(u_1)p(\sigma_1) \cdots p(\sigma_p)p(u_{p+1})\right) = \hat{M}_i p(s)$. □

Lemma 2. *Assume $\Sigma_{i,\mathrm{o}} \subseteq \hat{\Sigma}$ for $i \in \mathcal{I}$. Then, $\hat{M}_i p(s) = M_i(s)$, for all $s \in \Sigma^*$.*

Proof. We write $s = u_1\sigma_1 \cdots u_p\sigma_p u_{p+1}$ with $\sigma_j \in \hat{\Sigma}$ for $j = 1, \ldots, p$ and $u_j \in (\Sigma - \hat{\Sigma})^*$ for $j = 1, \ldots, p+1$. Then, $\hat{M}_i p(s) = \hat{M}_i(\sigma_1 \cdots \sigma_p) = \hat{M}_i(\sigma_1) \cdots \hat{M}_i(\sigma_p)$. Also, $M_i(s) = M_i(\sigma_1) \cdots M_i(\sigma_p)$ since $M_i(u_j) = \epsilon$ for $j = 1, \ldots, p+1$. Furthermore, by definition, $M_i(\sigma_j) = \hat{M}_i(\sigma_j)$ for $j = 1, \ldots, p$. Hence, $\hat{M}_i p(s) = \hat{M}_i(\sigma_1) \cdots \hat{M}_i(\sigma_p) = M_i(\sigma_1) \cdots M_i(\sigma_p) = M_i(s)$. □

Now, Theorem 1 can be proved.

[3] In general, natural projections can lead to an exponential increase in the state space of the abstracted model (Wong, 1998).

Proof. "(1)": Let $\hat{K}$ be codiagnosable for $\hat{G}$ and $\hat{M}_i, i \in \mathcal{I}$, with the worst-case detection delay $\hat{n}$, i.e., for all $\hat{s} \in L(\hat{G}) - \hat{K}$ and $\hat{s}\hat{t} \in L(\hat{G})$ s.t. $|\hat{t}| > \hat{n}$ or $\hat{s}\hat{t}$ deadlocks, there is $i \in \mathcal{I}$ s.t. $\forall \hat{u}_i \in \hat{M}_i^{-1}\hat{M}_i(\hat{s}\hat{t}) \cap L(\hat{G}), \hat{u}_i \notin \hat{K}$.

We want to show that $K = \hat{K} \parallel L(G)$ is codiagnosable for G and $M_i, i \in \mathcal{I}$. Assume the contrary, i.e., $\forall n \in \mathbb{N}$, there is $s \in L(G) - K$ and $st \in L(G)$ s.t. $|t| > n$ or st deadlocks but for all $i \in \mathcal{I}$, there exists a $u_i \in M_i^{-1}M_i(st) \cap L(G)$ s.t. $u_i \in K$. In particular, let $n = N \cdot \hat{n}$. Then, $\hat{s} := p(s) \in L(\hat{G}) - \hat{K}, \hat{s}\hat{t} := \hat{s}p(t) \in L(\hat{G})$ and $|\hat{t}| > \hat{n}$ or $\hat{s}\hat{t}$ deadlocks since p is a loop-preserving observer with bound N. Furthermore, using Lemma 1 and basic properties of natural projections, it holds for all $i \in \mathcal{I}$ that $\hat{u}_i := p(u_i) \in p(M_i^{-1}M_i(st) \cap L(G)) \subseteq p(M_i^{-1}\hat{p}_i^{-1}\hat{p}_i M_i(st)) \cap L(\hat{G}) = p(p^{-1}\hat{M}_i^{-1}\hat{M}_i p(st)) \cap L(\hat{G}) = \hat{M}_i^{-1}\hat{M}_i(\hat{s}\hat{t}) \cap L(\hat{G})$. In addition, $\hat{u}_i \in \hat{K}$ since $u_i \in K$, which contradicts that $\hat{K}$ is codiagnosable for $\hat{G}$ and $\hat{M}_i, i \in \mathcal{I}$.

To address the state space reduction, we note that it is shown in Wong (1998, Theorem 3.1.1) that the abstraction $\hat{G}$ cannot have a larger state space than the original model G if the projection p is an observer.

"(2)": Assume that K is codiagnosable for G and $M_i, i \in \mathcal{I}$, with the worst-case detection delay n, i.e., $\forall s \in L(G) - K$ and $st \in L(G)$ s.t. $|t| > n$ or st deadlocks there is $i \in \mathcal{I}$ s.t. for all $u \in M_i^{-1}M_i(st), u \notin K$.

We want to show that $\hat{K}$ is codiagnosable for $\hat{G}$ and $\hat{M}_i, i \in \mathcal{I}$. Assume the contrary, i.e., $\forall \hat{n} \in \mathbb{N}$, there is a $\hat{s} \in L(\hat{G}) - \hat{K}$ and $\hat{s}\hat{t} \in L(\hat{G})$ s.t. $|\hat{t}| > \hat{n}$ or $\hat{s}\hat{t}$ deadlocks but for all $i \in \mathcal{I}$, there exists a $\hat{u}_i \in \hat{M}_i^{-1}\hat{M}_i(\hat{s}\hat{t}) \cap L(\hat{G})$ s.t. $\hat{u}_i \in \hat{K}$. In particular, let $\hat{n} = n$. Then, since p is a loop-preserving observer, there are $s \in p^{-1}(\hat{s}), st \in p^{-1}(\hat{s}\hat{t})$ s.t. $s \in L(G) - K$ and $st \in L(G)$ and $|t| > n$ of st deadlocks. Considering that $\hat{u}_i \in \hat{M}_i^{-1}\hat{M}_i(\hat{s}\hat{t}) \cap L(\hat{G})$, there is $u_i \in p^{-1}(\hat{u}_i)$ s.t. $u_i \in L(G)$. Now, Lemma 2 implies that also $u_i \in p^{-1}(\hat{u}_i) \subseteq p^{-1}\hat{M}_i^{-1}\hat{M}_i(\hat{s}\hat{t}) = p^{-1}\hat{M}_i^{-1}\hat{M}_i p(st) = M_i^{-1}M_i(st)$, i.e., $u_i \in M_i^{-1}M_i(st) \cap L(G)$. Furthermore, $u_i \in K = \hat{K} \parallel L(G)$ since $\hat{u}_i \in \hat{K}$, which contradicts the assumption that K is codiagnosable for G and $M_i, i \in \mathcal{I}$. □

The result in Theorem 1 allows one to verify codiagnosability by applying the algorithm in Qiu and Kumar (2006) to the abstracted entities $\hat{G}$, $\hat{K}$ and $\hat{\Sigma}$. However, it is still required to enumerate the overall system G in order to compute the abstraction $\hat{G}$, which is not feasible for large-scale DES. In the next section, we consider a practical situation, where the explicit computation of G can be avoided.

Remark 1. Regarding the choice of $\hat{\Sigma}$, it is required that $\Sigma' \subseteq \hat{\Sigma} \subseteq \Sigma$ while meeting the sufficient conditions in Theorem 1, whereas it is desired that $\hat{\Sigma}$ contains as few events as possible in order to obtain a small abstracted model. In Schmidt (in press), we develop a polynomial-time algorithm that extends the initial alphabet Σ' until an appropriate $\hat{\Sigma}$ is found. It is implemented in the `libFAUDES` software library for DES (LibFAUDES, 2006–2010).

4. Codiagnosability for composed systems

In Zhou et al. (2008), DES that are modeled as the composition of multiple subsystems with their own observations are investigated. It is shown that decentralized diagnosers without communication can be computed using the subsystem models if the condition of *modular diagnosability* is fulfilled. However, the overall model has to be evaluated to verify this condition. In this section, we apply the abstraction-based codiagnosability verification to the scenario in Zhou et al. (2008). We show that additional computational savings can be achieved by avoiding the enumeration of the overall system model.

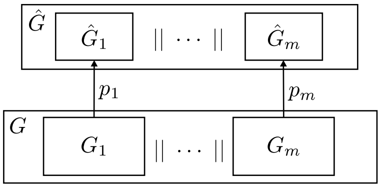

Fig. 2. Model abstraction for composed systems.

4.1. Abstraction-based verification

We consider models G that are composed of m subsystems G_i over the alphabets $\Sigma_i, i \in \mathcal{I}$ s.t. $G = \|_{i \in \mathcal{I}} G_i$ and $\Sigma = \bigcup_{i \in \mathcal{I}} \Sigma_i$. Assuming that each subsystem obtains its own observations, the observation masks are defined as $M_i : \Sigma_i \to \Delta_i \cup \{\epsilon\}, i \in \mathcal{I}$. In addition, the correct system behavior is again given by a specification $K \subseteq L(G)$. It is shown in Zhou et al. (2008) that modular diagnosability holds if K is codiagnosable for G and $M_i\theta_i, i \in \mathcal{I}$ with the natural projections $\theta_i : \Sigma^* \to \Sigma_i^*$.

We cite two results that enable the abstraction-based codiagnosability verification for composed systems.

Proposition 1. *Let $G_i, i \in \mathcal{I}$ be given as above and define $\Sigma_{i,\cap} := \bigcup_{j \in \mathcal{I}, j \neq i} (\Sigma_i \cap \Sigma_j)$ as the set of* sharedevents *for each $i \in \mathcal{I}$. Then the following holds for $p_i : \Sigma_i^* \to \hat{\Sigma}_i^*$ with $\Sigma_{i,\cap} \subseteq \hat{\Sigma}_i \subseteq \Sigma_i, i \in \mathcal{I}$ and $\hat{\Sigma} := \bigcup_{i \in \mathcal{I}} \hat{\Sigma}_i$.*

(1) *$L(\hat{G}) = p(L(G)) = \|_{i \in \mathcal{I}} p_i(L(G_i)) = \|_{i \in \mathcal{I}} L(\hat{G}_i)$ (Wonham, 2008, Exercise 3.3.7).*
(2) *If p_i is a loop-preserving observer for $L(G_i)$ for all $i \in \mathcal{I}$, then also $p : \Sigma^* \to \hat{\Sigma}^*$ is a loop-preserving observer for $L(G)$ (Schmidt, 2010).*

Hence, as depicted in Fig. 2, it is possible to obtain $\hat{G}$ by composing the abstracted subsystems $\hat{G}_i$ with $L(\hat{G}_i) = p_i(L(G_i))$. Using the abstracted observation mask $\hat{M}_i : \hat{\Sigma}_i \to \hat{\Delta}_i \cup \{\epsilon\}$ and defining $\hat{\theta}_i : \hat{\Sigma}^* \to \hat{\Sigma}_i^*$, codiagnosability can be verified based on $\hat{G}$ and observations through $\hat{M}_i\hat{\theta}_i$.

Theorem 2. *Assume the notation in this section.*

(1) *Problem 1(1) is solved if 1. $\Sigma_{i,\cap} \subseteq \hat{\Sigma}_i$ for all $i \in \mathcal{I}$, 2. p_i is a loop-preserving observer for all $i \in \mathcal{I}$, 3. M_i is consistent with p_i for all $i \in \mathcal{I}$.*
(2) *Problem 1(2) is solved if additionally $\Sigma_{i,\mathrm{o}} = \{\sigma \in \Sigma_i | M_i(\sigma) \neq \epsilon\} \subseteq \hat{\Sigma}_i$ for all $i \in \mathcal{I}$.*

Proof. We show that the conditions in Theorem 1 are fulfilled. Because of Proposition 1, p is a loop-preserving observer. Furthermore, since M_i is consistent with p_i, also $M_i\theta_i$ is consistent with p for all $i \in \mathcal{I}$.

Finally, $\Sigma_{\mathrm{o}} = \bigcup_{i \in \mathcal{I}} \Sigma_{i,\mathrm{o}} \subseteq \hat{\Sigma}$ since $\Sigma_{i,\mathrm{o}} \subseteq \hat{\Sigma}_i$ for $i \in \mathcal{I}$. □

Remark 2. Note that the same procedure as referred to in Remark 1 can be used in order to find appropriate abstraction alphabets $\hat{\Sigma}_i$ for $i \in \mathcal{I}$. Here, the initial alphabet for each $i \in \mathcal{I}$ is $\Sigma_{i,\cap} \cup (\Sigma_i \cap \Sigma')$.

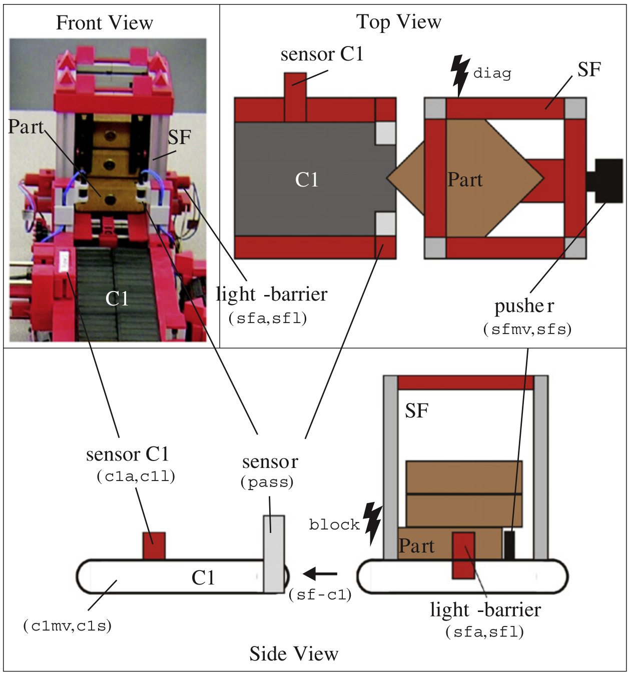

Fig. 3. Stack feeder (SF) and conveyor (C1).

4.2. Application example

We illustrate the method in the previous section by the manufacturing unit in Fig. 3. It consists of a *stack feeder* (SF) that can transport *parts* to the *conveyor belt* C1.

A light-barrier at SF is used to detect if a part arrives (event sfa) or leaves (sfl). If a part is present, a small pusher that is mounted to a belt can push the part toward C1 (sfmv) until the belt stops (sfs). Furthermore, the *shared events* sf-c1 and pass with C1 indicate that the operation of the unit is initiated and that a part passes the sensor between SF and C1, respectively. The subautomaton of G_{SF} in Fig. 4 that has states with a white background describes the closed-loop behavior of the SF in analogy to a supervisor synthesis in Schmidt et al. (2008).

C1 moves (c1mv) after the occurrence of sf-c1 and can both detect if a part arrives from SF (pass) and reaches the sensor at C1 (c1a). Then, C1 stops (c1s), and a new transport can start if the part is removed from C1 (c1l). This behavior is characterized by G_{C1} in Fig. 4.

In addition, we include two potential faults in our model. First, it is possible that the transport of a present part at SF is blocked (unobservable event block) such that the part cannot leave the light-barrier although the pusher tries to move it. The faulty behavior appears in G_{SF} and G_{C1}, where a timer elapses when the expected part does not pass the sensor between C1 and SF (states shaded in light gray). Second, a part can be placed with a wrong orientation (unobservable event diag) as shown by the top view in Fig. 3. In that case, while moving, the pusher lifts the part and puts it down again such that the events sfl and sfa are observed. However, since the part cannot leave the SF, the event pass does not occur such that the belt does not stop (dark gray states in G_{SF}).

Our goal is now to verify if the faulty behavior can be diagnosed by two local agents that observe the behavior of SF and C1, respectively. To this end, we introduce the reduced specification $K' = \{\epsilon\}$ over the alphabet $\Sigma' = \{\texttt{block}, \texttt{diag}\}$ that disallows the occurrence of block and diag. Furthermore, the observation masks M_{SF} and M_{C1} are described by $M_{SF}(\texttt{block}) = M_{SF}(\texttt{diag}) = M_{C1}(\texttt{block}) = \epsilon$, while $M_{SF}(\sigma_{SF}) = \sigma_{SF}$ and $M_{C1}(\sigma_{C1}) = \sigma_{C1}$ for all remaining $\sigma_{SF} \in \Sigma_{SF} - \{\texttt{block}, \texttt{diag}\}$ and $\sigma_{C1} \in \Sigma_{C1} - \{\texttt{block}\}$, respectively.

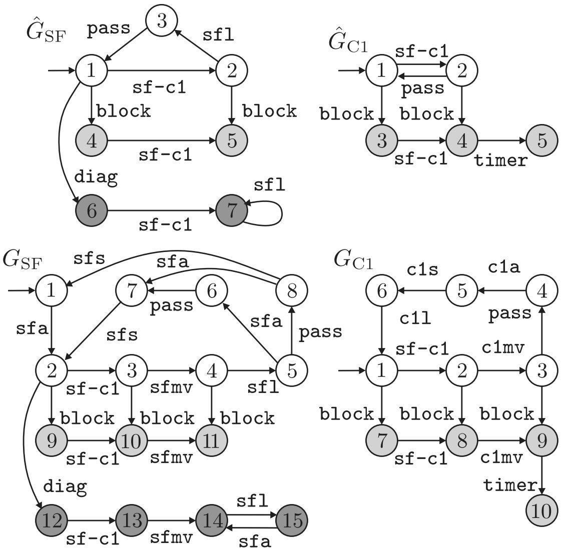

Fig. 4. Subsystem models and abstractions for SF and C1.

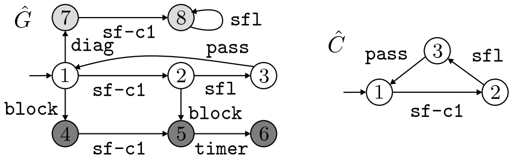

Fig. 5. Abstracted model $\hat{G}$ and specification $\hat{K}$.

We determine the abstraction alphabets $\hat{\Sigma}_{SF} = \{\texttt{sf-c1}, \texttt{pass}, \texttt{sfl}, \texttt{block}, \texttt{diag}\}$ and $\hat{\Sigma}_{C1} = \{\texttt{sf-c1}, \texttt{pass}, \texttt{block}, \texttt{timer}\}$ as pointed out in Remark 2 to compute the abstracted models $\hat{G}_{SF}$ and $\hat{G}_{C1}$ in Fig. 4. Since all conditions in Theorem 2(1) are fulfilled, we employ the abstracted model $\hat{G} = \hat{G}_{SF} \parallel \hat{G}_{C1}$ and the abstracted specification $\hat{K} = L(\hat{C})$ in Fig. 5 for the abstraction-based codiagnosability verification. On the one hand, it holds that the faulty string block can be uniquely distinguished from correct strings after the extended string block sf-c1 timer by the local site C1. On the other hand, the occurrence of diag is uniquely identified as a fault after the extension diag sf-c1 sfl sfl by SF. Hence, it can be concluded from the codiagnosability of $\hat{K}$ for $\hat{G}$ and $\hat{M}_{SF}$, $\hat{M}_{C1}$ that also the original specification $K = K' \parallel L(G)$ is codiagnosable for $G = G_{SF} \parallel G_{C1}$ and M_{SF}, M_{C1}.

Together, the abstraction-based codiagnosability verification can be carried out with the abstracted model $\hat{G}$ with 8 states and the abstracted specification $\hat{K}$ with 3 states. In contrast, the original verification would necessitate the evaluation of the model G with 41 states and the specification K with 26 states, which illustrates the computational savings of the abstraction-based method.

5. Conclusion

In this paper, an efficient approach for the verification of *codiagnosability* for the decentralized diagnosis without communication of composed discrete event systems was proposed. Instead of using the overall system model, our method allows one to perform the codiagnosability test based on an *abstracted model* on a smaller state space. In addition, it was shown that further computational savings can be achieved in the practical case of large-scale discrete event systems that are composed of multiple subsystems. Thus, it is possible to apply the abstraction to the small subsystem models, rather than the large overall system model. The benefits of our method were demonstrated by a small manufacturing system example with two subsystems.

References

Cassandras, C. G., & Lafortune, S. (2006). *Introduction to discrete event systems*. Secaucus, NJ, USA: Springer-Verlag New York, Inc..

Debouk, R., Malik, R., & Brandin, B. (2002). A modular architecture for diagnosis of discrete event systems. In *Decision and control, conference on*.

Debouk, R., & Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, *10*, 33–86.

García, E., Correcher, A., Morant, F., Quiles, E., & Blasco, R. (2005). Modular fault diagnosis based on discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, *15*(3), 237–256.

Hashtrudi Zad, S., Kwong, R., & Wonham, W. (2003). Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Transactions on Automatic Control*, *48*(7), 1199–1212.

LibFAUDES (2006–2010). LibFAUDES software library for discrete event systems. URL: www.rt.eei.uni-erlangen.de/FGdes/faudes.

Paoli, A., & Lafortune, S. (2005). Safe diagnosability for fault-tolerant supervision of discrete-event systems. *Automatica*, *41*(8), 1335–1347.

Qiu, W., & Kumar, R. (2006). Decentralized failure diagnosis of discrete event systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *36*(2), 384–395.

Qiu, W., & Kumar, R. (2008). Distributed diagnosis under bounded-delay communication of immediately forwarded local observations. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *38*(3), 628–643.

Qiu, W., & Kumar, R. (2004). Decentralized failure diagnosis of discrete event systems. In *Discrete event systems, workshop on*.

Ricker, S. L., & van Schuppen, J. H. (2001). Decentralized failure diagnosis with asynchronous communication between supervisors. In *European control conference*.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*(9), 1555–1575.

Schmidt, K. (2010). Abstraction-based failure diagnosis for discrete event systems. *System & Control Letters*, *59*, 42–47.

Schmidt, K. (2010). Computation of projections for the abstraction-based diagnosability verification. In *Discrete event systems, workshop on* (in press).

Schmidt, K., Moor, T., & Perk, S. (2008). Nonblocking hierarchical control of decentralized discrete event systems. *IEEE Transactions on Automatic Control*, *53*(10), 2252–2265.

Su, R., & Wonham, W. (2005). Global and local consistencies in distributed fault diagnosis for discrete-event systems. *IEEE Transactions on Automatic Control*, *50*(12), 1923–1935.

Takai, S. (2008). A sufficient condition for diagnosability of large-scale discrete event systems. In *Technical conference on circuits/systems, computers and communications*.

Wang, Y., Yoo, T.-S., & Lafortune, S. (2007). Diagnosis of discrete event systems using decentralized architectures. *Discrete Event Dynamic Systems: Theory and Applications*, *17*(2), 233–263.

Wong, K. C. (1998). On the complexity of projections of discrete-event systems. In *Discrete event systems, workshop on* (pp. 201–208).

Wong, K. C., & Wonham, W. M. (1996). Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, *6*(3), 241–273.

Wonham, W. M. (2008). Supervisory control of discrete-event systems. Department of Electrical and Computer Engineering, University of Toronto. URL: http://www.control.utoronto.ca/DES.

Yoo, T.-S., & Garcia, H. E. (2008). Diagnosis of behaviors of interest in partially-observed discrete-event systems. *System & Control Letters*, *57*(12), 1023–1029.

Zhou, C., Kumar, R., & Sreenivas, R. (2008). Decentralized modular diagnosis of concurrent discrete event systems. In *Discrete event systems, workshop on* (pp. 388–393).

K. Schmidt received the Diploma and Ph.D. degrees from the University of Erlangen-Nuremberg, Germany, in 2002 and 2005, respectively, both in Electrical, Electronic, and Communication Engineering.

He is currently an Assistant Professor at the Department of Electronic and Communication Engineering, Cankaya University, Ankara. His research interests include controller synthesis and failure diagnosis for discrete event systems, industrial automation systems, and vehicular communication networks.