Discrete Event Dyn Syst (2006) 16: 9–37
DOI: 10.1007/s10626-006-6177-5

RESEARCH ARTICLE

Diagnosability of Discrete Event Systems with Modular Structure

Olivier Contant · Stéphane Lafortune · Demosthenis Teneketzis

Abstract The diagnosis of unobservable faults in large and complex discrete event systems modeled by parallel composition of automata is considered. A modular approach is developed for diagnosing such systems. The notion of modular diagnosability is introduced and the corresponding necessary and sufficient conditions to ensure it are presented. The verification of modular diagnosability is performed by a new algorithm that incrementally exploits the modular structure of the system to save on computational effort. The correctness of the algorithm is proved. Online diagnosis of modularly diagnosable systems is achieved using only local diagnosers.

Keywords Distributed systems · Diagnosability · Modularity · Common events

Introduction

Many application areas including document processing systems, heating, ventilation, and air-conditioning systems, intelligent transportation systems, chemical process control, and telecommunication networks have successfully implemented monitoring and diagnosis methodologies based upon discrete-event models of dynamic systems. Most discrete-event fault diagnosis methodologies necessitate the construction of a "monolithic" model of the system under consideration for diagnosability analysis and implementation; see Aghasaryan et al. (1998), Bouloutas et al. (1994), Console et al. (2002), Contant et al. (2004b), Debouk et al. (2000), Hashtrudi Zad et al. (2003), Jiang and Kumar (2002), Jiang et al. (2003), Lafortune et al. (2001), Lamperti and Zanella (2003, 2004); Lin (1994), Lunze (2000), Pencolé et al. (2002), Sampath (2001), Sampath et al. (1995, 1996, 1998), Sengupta (2001), Sinnamohideen (2001), Yoo and Garcia (2004).

Almost all systems possess a "modular" structure. In general a modular system is composed of several modules, local components, or subsystems that could themselves possibly be formed of several smaller individual modules. Such modular

O. Contant (✉) · S. Lafortune · D. Teneketzis
Department of Electrical Engineering and Computer Science,
The University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109-2122, USA
*Olivier Contant is now working at Microsoft Corporation.
e-mail: Olivier.Contant@microsoft.com

systems tend to have very large state spaces and thus they are complex to model and difficult to diagnose in a holistic manner. Therefore, diagnosis methodologies that take advantage of the natural decomposition of a modular system have the potential to provide conceptual and computational advantages as compared to methodologies requiring a monolithic representation of the entire system. Individual modules are in general simpler to diagnose locally, but such diagnosis may not account for the effect of the rest of the system. Therein lies the challenge of modular diagnosis methodologies: performing diagnosis locally, i.e., at each module, while at the same time accounting for the coupling of each module with the rest of the system. Approaches that exploit the modular structure of a system for monitoring and diagnosis have been developed in Benveniste et al. (2003), Debouk et al. (2002), Garcia et al. (2002), Genc and Lafortune (2003), Holloway and Chand (1994), Pandalai and Holloway (2000), Pencolé (2000), Ricker and Fabre (2000), Su and Wonham (2004), Su et al. (2002). This paper has an objective similar to the one in the previously mentioned references; however, the approach adopted here is different. We review some relevant references below.

In Debouk et al. (2002), the authors implement a modular architecture for local diagnosis at each component. The procedure requires the construction of a local diagnoser for each component. Sufficient conditions that ensure global diagnosability are given. Diagnosability is verified using an incremental approach that checks individually each subsystem. While our setup is similar to that in Debouk et al. (2002), we use a different notion of diagnosability and further exploit the structure of the system in verifying it. In Pencolé (2000), the authors are interested in diagnosing distributed systems. Each system is composed of spatially distributed subsystems. Each subsystem runs a local diagnoser, i.e., diagnoses only faults occurring at the site. The complete system diagnosis is obtained by suitably merging the local information of all subsystems, i.e., local models and sequences of observable events. In Su et al. (2002), the authors develop a distributed diagnosis method with communication. Local diagnosers are built for each local component. Communication is possible via input and output connections that link the local diagnosers. Local diagnosis is based mainly on local observations, and communication is used for refinement purposes. Our approach is different from Pencolé (2000), Su et al. (2002) since we are not assuming real time communication among local diagnosers. In Ricker and Fabre (2000), the authors present a modular approach for detecting faults in large and distributed systems. The subsystems interact among one another via sets of common resources that are assumed observable. However the problem formulation in Ricker and Fabre (2000) is different from ours. In Holloway and Chand (1994), the authors develop a methodology to monitor distributed faults in manufacturing systems. Unlike our approach which is based on untimed models, the methodology in Holloway and Chand (1994) employs timed models (called "time templates"). The monitoring is done by a set of distributed interconnected processors and event prediction is made possible by the use of sets of timing and sequencing relationships available through the different templates assigned to processors.

The approach proposed in this paper is suited to systems that are modeled by the parallel composition of automata, where each automaton represents a local component, subsystem, or module of the global system. Furthermore, the proposed approach focuses on the case where building a monolithic model of the complete system is impractical for computational or other reasons. In this context, the cen-

tralized and decentralized versions of the so-called Diagnoser Approach in Sampath et al. (1995) and Debouk et al. (2000) are inappropriate. In order to possibly alleviate the drawbacks of methodologies based on monolithic models of the entire system, it is essential to exploit the modular structure of the system, which naturally emanates from the local automata that are coupled by parallel composition.

The first objective of the approach investigated in this paper is the on-line diagnosis of a modular system using solely the diagnosers corresponding to individual modules, or local diagnosers. In this regard, we define the property of "modular diagnosability." It is a variation of the definition of diagnosability in (Sampath et al., 1995) that accounts for the modular structure of the system. We show in the paper that systems that are modularly diagnosable can be diagnosed by employing only local diagnosers. The second objective is to verify in a manner as efficient as possible whether or not modular diagnosability holds. For this purpose, we present an algorithm that is incremental in nature and that attempts to verify modular diagnosability by considering one module at a time and without necessarily using all the remaining modules in the system. Roughly speaking, the idea is to use only the information about other modules that is absolutely necessary in determining what faults can be diagnosed with certainty by a local diagnoser.

The contributions of this paper and its organization are as follows. The System Model section presents some necessary notation used extensively throughout this paper. The first contribution of the paper is the introduction of the notion of *modular diagnosability*, which is explicitly geared towards systems with modular representations. This notion is presented in the Modular Diagnosability section. The main feature of modular diagnosability is that it requires persistent excitation of the module or local component that exhibits a faulty behavior. The motivation and intuition of the notion of modular diagnosability are presented and discussed along with associated necessary and sufficient conditions. The Preliminary Discussion section gives a preliminary discussion on the approach developed in the following sections for testing modular diagnosability. The Properties of Modular Diagnosability section presents properties of modular diagnosability that constitute the foundations of the correctness proof of the developed approach. The second main contribution of this paper is the development in the Test for Modular Diagnosability section of a novel algorithm for verifying modular diagnosability. This algorithm (abbreviated as MDA hereafter) proceeds incrementally by considering the automata models of other system components only if they are required to draw definitive conclusions about the diagnosability of faults within a given system component. This section also contains the correctness proof of MDA and two examples that illustrate its application. Finally, online diagnosis based solely on local diagnosers is discussed in the Online Diagnosis section. A summary of contributions and results concludes the paper in the Conclusion section.

System Model

We assume that the reader is familiar with basic notions[1] in languages and automata theory and with the notation and main concepts of the Diagnoser Approach

[1] See, e.g., Chapter 2 of Cassandras and Lafortune (1999).

introduced in Sampath et al. (1995) and used in several papers thereafter, such as diagnosers, certain and uncertain states, and indeterminate cycles.

Let I be the total number of components or individual subsystems in the given modular system under consideration. Let $T = \{1,\ldots, I\}$ and $S \subseteq T$. Elements i of T are often termed "sites" or "modules" hereafter. We use the notation

$$G_S = (X_S, \Sigma_S, \delta_S, x_{S_0}, X_{S_m}) \tag{1}$$

to denote the automaton with state space X_S, set of events Σ_S, (partial) transition function δ_S, initial state x_{S_0}, and set of marked states X_{S_m}. When $S = T$, G_S denotes the global (or complete) system model. When $S = \{i\}$, with $i \in T$, G_S denotes individual component model i. When $S \subset T$, $S \neq \{i\}$, $i \in T$, G_S denotes the partial system model (or subsystem) comprised of the individual automata in the set S, which we will call the "system G_S" hereafter. In all of the above cases, system G_S accounts for the normal and failed behavior of the components in S, consistent with the Diagnoser Approach of Sampath et al. (1995) and Sampath et al. (1996). $G_S = \|_{z\in S} G_z$ is obtained by composing the individual automata G_z, $z \in S$, using the parallel composition operation.

The behavior of the system G_S is described by the prefix-closed language $\mathcal{L}(G_S)$ generated by G_S. $\mathcal{L}(G_S)$ is assumed to be live. This means that there is a transition defined at each state x in X_S, i.e., G_S cannot reach a point at which no event is possible. The liveness assumption on $\mathcal{L}(G_S)$ is made for the sake of simplicity. With slight technical modifications, all the main results of this paper hold true when the liveness assumption is relaxed, cf. the approach employed in Sampath et al. (1998). Some of the events in Σ_S are observable, i.e., their occurrence can be observed by sensors, while the rest are unobservable. We use the notation Σ_{o_S} and Σ_{uo_S} to represent the set of observable and unobservable events of G_S, respectively, where $\Sigma_{o_S} = \Sigma_S \setminus \Sigma_{uo_S}$. Let Σ_{f_S} denote the set of fault events in the system G_S. Without loss of generality, we assume that $\Sigma_{f_S} \subseteq \Sigma_{uo_S}$, since an observable fault event can be diagnosed trivially.

It will be necessary in many instances to explicitly identify the event set associated with an automaton. By default, the event set Σ_S associated with automaton G_S will be $\Sigma_S := \{\sigma \in s : s \in \mathcal{L}(G_S)\}$, namely, all the events that appear in all the traces in $\mathcal{L}(G_S)$. Similarly, $\Sigma_z = \{\sigma \in s : s \in \mathcal{L}(G_z)\}$. Hence, in general, $\Sigma_S \subseteq \bigcup_{z\in S} \Sigma_z$, due to the effect of parallel composition. In special cases it is required to define a larger set of events associated with G_S than the default one. In this regard, the notation (G_S, Σ) will denote the automaton G_S together with the event set Σ such that $\Sigma \supseteq \Sigma_S$. For example, (G_i, Σ_S) implies that the original event set Σ_i of G_i is now augmented by the set $\Sigma_S \backslash \Sigma_i$. While (G_i, Σ_S) has the same language properties as (G_i, Σ_i), it has different behavior when performing parallel composition (or, more generally, any operation using sets of events) with other modules as it will prevent the occurrence of events in $\Sigma_S \backslash \Sigma_i$.

We use the notation $\Sigma = \Sigma' \dot{\cup} \Sigma''$ to indicate that the event set Σ is the disjoint union of Σ' and Σ'', i.e., $\Sigma' = \Sigma \setminus \Sigma''$. Let $R \subset T$. The event set Σ_R of subsystem G_R is partitioned as $\Sigma_R = \Sigma_{CM_R} \dot{\cup} \Sigma_{PV_R}$, where $\Sigma_{CM_R} = \Sigma_R \cap [\cup_{z\in T\setminus R} \Sigma_z]$ represents the set of common events in G_R and Σ_{PV_R} represents the set of private events in G_R. We define $\Sigma_{CM} = \cup_{i\in T} \Sigma_{CM_i}$ the set of *all* common events, namely, the set of all events that are common to two or more individual components. We use the notation $\Sigma_{CM_{o_R}} = \Sigma_{CM_R} \cap \Sigma_{o_R}$ to represent the set of common and observable events of subsystem R.

The notation $CoAc(G_S)$ represents the automaton obtained from G_S by retaining only those states x of G_S that are coaccessible, namely, that can reach a (marked) state in X_{S_m}. This notation will often be specialized to $CoAc(G_S, M_x)$ where M_x will be a particular label associated with the marked states of G_S. In this case, $CoAc(G_S, M_x)$ will be the coaccessible part of G_S with respect to the marked states of G_S that are labeled with M_x.

An observer[2] is a tool that allows to transform a non-deterministic automaton into a deterministic one. In the DES literature, an observer is used to estimate the states of the deterministic automaton G with "observable" events Σ_o, which is possible by considering unobservable events Σ_{uo} as empty events ε (and therefore making it non-deterministic). This notion is a natural one in the sense that the observer observes only observable events. In this paper we do not want to restrict its application only to observable events. We are interested in extending the notion of observer to any specific set of events Σ^{spe} (i.e., any set of special events of interest for the detection process, such as, common events, events of specific system components, etc.). Let Σ be the set of events of the deterministic automaton G and $\Sigma_c^{spe} = \Sigma \setminus \Sigma^{spe}$ be the complement of the set Σ w.r.t. Σ^{spe} (where "spe" stands for "special"). We extend the procedure presented in Cassandras and Lafortune (1999) to build an observer for a deterministic automaton with respect to special events Σ^{spe}. This is achieved by processing all events in Σ_c^{spe} as if they were ε. Therefore, the event set of the observer will be $\Sigma^{obs} := \Sigma \cap \Sigma^{spe}$ (where "obs" stands for "observer"). We define

$$Obs(G, \Sigma^{spe}) = (X^{obs}, \Sigma^{obs}, \delta^{obs}, x_0^{obs}, X_m^{obs}) \tag{2}$$

to be the state estimator of G with respect to Σ^{spe}. This construction provides a generic tool that we use in our algorithm in the Test for Modular Diagnosability section. For example, $Obs(G_i, \Sigma_{CM_l})$, $l \neq i$, is the state estimator of G_i with respect to Σ_{CM_l}, the common events of G_l.

Let Σ_X and Σ_Y be any sets of events. We define two projection operators relative to these sets, $P_{\{\Sigma_X,\Sigma_Y\}}$ for the usual natural projection and $R_{\{\Sigma_X,\Sigma_Y\}}$ for the so-called "reverse" projection. Specifically, the natural projection $P_{\{\Sigma_X,\Sigma_Y\}} : \Sigma_X^* \to \Sigma_Y^*$ is defined in the usual manner:

$$P_{\{\Sigma_X,\Sigma_Y\}}(\varepsilon) := \varepsilon \tag{3}$$

$$P_{\{\Sigma_X,\Sigma_Y\}}(\sigma) := \begin{cases} \sigma & \text{if } \sigma \in \Sigma_y \\ \varepsilon & \text{if } \sigma \notin \Sigma_Y \end{cases} \tag{4}$$

$$P_{\{\Sigma_X,\Sigma_Y\}}(s\sigma) := P_{\{\Sigma_X,\Sigma_Y\}}(s) P_{\{\Sigma_X,\Sigma_Y\}}(\sigma) \quad \text{for } s \in \Sigma_X^*, \sigma \in \Sigma_X. \tag{5}$$

In contrast, the *reverse projection* $R_{\{\Sigma_X,\Sigma_Y\}}$ is applied to traces of events from Σ_X^* and produces "inverse" traces from Σ_Y^* as is usually done in inverse projection operations. More precisely, $R_{\{\Sigma_X,\Sigma_Y\}} : \Sigma_X^* \to 2^{\Sigma_Y^*}$ is defined as follows:

$$R_{\{\Sigma_X,\Sigma_Y\}}(s) = \left\{ t \in \Sigma_Y^* : P_{\{\Sigma_Y,\Sigma_X\}}(t) = s \right\}. \tag{6}$$

[2] See Cassandras and Lafortune (1999) for the basic definition of an observer.

The natural and reverse projections can also be defined with respect to a particular language L. For $L \subseteq \Sigma_Y^*$,

$$P^L_{\{\Sigma_X,\Sigma_Y\}}(s) = \{t \in L : P_{\{\Sigma_X,\Sigma_Y\}}(s) = t\}, \tag{7}$$

and

$$R^L_{\{\Sigma_X,\Sigma_Y\}}(s) = \{t \in L : P_{\{\Sigma_Y,\Sigma_X\}}(t) = s\}. \tag{8}$$

We conclude this section by stating a key assumption that will be required for the results presented in the remainder of the paper.

ASSUMPTION 1 *Observability of Common Events*

For each module $i \in T$, the common events of module i are observable, namely, $\Sigma_{CM_i} \subseteq \Sigma_{o_i}$.

This assumption implies that all faults are private events (since they are unobservable). We leave as future work the development of diagnostic methodologies when: (i) common events are not observable; and (ii) all common events belonging to one module are not observable by that module, but each common event is observable by at least one module.

Modular Diagnosability

We define the notion of modular diagnosability as follows. Given a set A the notation $(\Sigma_a : a \in A)$ represents the list of sets Σ_a for all elements $a \in A$. For the sake of generality, the definition is given with two parameter sets, S^- and S, that are such that $S^- \subseteq S \subseteq T$.

DEFINITION 1 *Modular Diagnosability*

Let $T = \{1, \ldots, I\}$, $S \subseteq T$, $G_S = \|_{z\in S} G_z$, and $S^- \subseteq S$. The language $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S^-)$ if $\forall i \in S^-$, $\forall f \in \Sigma_{f_i}$, $\forall s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\exists n \in \mathbb{N}$ s.t. $\forall t \in \mathcal{L}(G_S)/s, \| P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \| \geq n \Rightarrow D(st) = 1$. The diagnosability condition function D is given by

$$D(st) = \begin{cases} 1 & \text{if } [\omega \in R^{\mathcal{L}(G_S)}_{\{\Sigma_{o_S},\Sigma_S\}}[P_{\{\Sigma_S,\Sigma_{o_S}\}}(st)] \Rightarrow f \in \omega], \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

To draw comparisons between our modular diagnosability algorithm, presented in the Test for Modular Diagnosability section, and any other potential approach, we give necessary and sufficient conditions for modular diagnosability based on monolithic constructions of the system model and diagnoser for a given set S of system components. Let $G_{d_S} = (Q_{d_S}, \Sigma_{o_S}, \delta_{d_S}, q_{0_S})$ and $G'_S = (X_{o_S}, \Sigma_{o_S}, \delta_{G'_S}, x_0)$ denote, respectively, the diagnoser of G_S and the non-deterministic automaton built from G_S by eliminating unobservable events. Both are defined in Sampath et al. (1995); these definitions are recalled in Appendix for the benefit of the readers.

Consider module $i \in T$ and diagnoser G_{d_S} where $i \in S$, $S \subseteq T$. We have the following definition.

DEFINITION 2 *F^{M_i}-indeterminate cycle in G_{d_S}*

A set of F-uncertain states $q_1, q_2, \ldots, q_n \in Q_{d_S}$ is said to form an F^{M_i}-indeterminate cycle in G_{d_S} if the following condition C1 is satisfied.

C1) States $q_1, q_2, \ldots, q_{\mathrm{n}} \in Q_{d_S}$ form a cycle in G_{d_S} with $\delta_{d_S}(q_u, \sigma_u) = q_{u+1}, u = 1, \ldots, n-1, \delta_{d_S}(q_n, \sigma_n) = q_1$ where $\sigma_u \in \Sigma_{o_S}$, $u = 1, \ldots, n$ and $\exists l \in \{1, \ldots, n\}$ s.t. $\sigma_l \in \Sigma_{\mathrm{o}_i}$.

Considering the states $q_1, q_2, \ldots, q_n \in Q_{d_S}$, $\exists (x_u^k, \ell_u^k)$, $(y_u^r, \widetilde{\ell}_u^r) \in q_u$, $u = 1, \ldots, n$, $k = 1, \ldots, m$, and $r = 1, \ldots, m'$ such that:

i) $[(F \in \ell_u^k) \wedge (F \notin \widetilde{\ell}_u^r)]$, for all u, k, and r, where F represents the label associated with the fault event $f \in \Sigma_{f_i}$, $i \in S$,

ii) the sequences of states x_u^k, $u = 1, \ldots, n$, $k = 1, \ldots, m$, and $\{y_u^r\}$, $u = 1, \ldots, n$, $r = 1, \ldots, m'$, form cycles in G_S' with

- $(x_u^k, \sigma_u, x_{(u+1)}^k) \in \delta_{G_S'}, u = 1, \ldots, n-1, k = 1, \ldots, m, (x_n^k, \sigma_n, x_1^{k+1}) \in \delta_{G_S'}, k = 1, \ldots, m-1, (x_n^m, \sigma_n, x_1^1) \in \delta_{G_S'}$, *and*
- $(y_u^r, \sigma_u, y_{(u+1)}^r) \in \delta_{G_S'}, \{u = 1, \ldots, n-1, r = 1, \ldots, m', (y_n^r, \sigma_n, y_1^{r+1}) \in \delta_{G_S'}, r = 1, \ldots, m'-1, (y_n^{m'}, \sigma_n, y_1^1) \in \delta_{G_S'}$.

Remark 1: The symbol "M_i," in the notation "F^{M_i}-indeterminate cycle," stands for "Module G_i." F^{M_i}-indeterminate cycles differ slightly from the F-indeterminate cycles introduced in Sampath et al. (1995) in two respects. First, we require that there exists at least one observable event from module $\mathrm{G_i}$ in the cycle of states $q_1, q_2, \ldots, q_n \in Q_{d_S}$: *cf.* "$\exists l \in \{1, \ldots, n\}$ s.t. $\sigma_l \in \Sigma_{o_i}$" in Condition C1. Second, we require that the label F in hypothesis (*i*) of Condition C1 represents the label associated with the fault event $f \in \Sigma_{f_i}$ i.e., the fault event f originates from module G_i.

THEOREM 1 *Consider the language $\mathcal{L}(G_S)$ generated by automaton $G_S = \|_{z \in S}\, G_z$. $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_i} : i \in S)$, if there are no F^{M_i}-indeterminate cycles in the diagnoser G_{d_S}, for all $i \in S$.*

The proof of Theorem 1 is similar to the one in Sampath et al. (1995) and is therefore omitted.

To gain insight into the definition of modular diagnosability, we reformulate with minor modifications the notion of diagnosability introduced in Sampath et al. (1995) and refer to it from now on as *monolithic diagnosability*.

DEFINITION 3 *Monolithic Diagnosability*

Let $T = \{1, \ldots, I\}$, $S \subseteq T$, and $G_S = \|_{z \in S}\, G_z$. The language $\mathcal{L}(G_S)$is monolithically diagnosable w.r.t. $(\Sigma_{\mathrm{o}_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$ if $\forall i \in S$, $\forall f \in \Sigma_{f_i}$, $\forall s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\exists n \in \mathbb{N}$ s.t. $\forall t \in \mathcal{L}(G_S)/s, \| P_{\{\Sigma_S, \Sigma_{\mathrm{o}_S}\}}(t) \| \geq n \Rightarrow D(st) = 1$, where the diagnosability condition function D is as in equation (9).

Remark 2: Definition 3 differs from the diagnosability definition introduced in Sampath et al. (1995) as follows:

i) We use the equation $\| P_{\{\Sigma_S,\Sigma_{o_S}\}}(t) \| \geq n$ instead of $\|t\| \geq n$. This modification implies that cycles of unobservable events are not taken into account when verifying the diagnosability properties of a system.
ii) The order of the quantifiers allows one natural number n for each trace s that ends with a fault event, instead of requiring one natural number for each fault event f, i.e., for all traces s ending with f. This change[3] allows for more precise choices of lower bounds for fault detection and identification.

In the special case where the considered system G_S is composed of a single module, i.e., $|S| = 1$, there are no differences between first, the modular and monolithic definitions, and second, F^{M_i}- and F-indeterminate cycles. In the general case, the main difference between the modular and the monolithic definitions of diagnosability concerns the type of traces that need to be considered. When testing for diagnosability of a fault event f at the end of trace s, we consider projections of any continuation t of length greater than n. For monolithic diagnosability, the projection of t is with respect to the observable events of system G_S, i.e., $\| P_{\{\Sigma_S,\Sigma_{o_S}\}}(t) \| \geq n$. For modular diagnosability, the projection of t is with respect to the observable events of system G_i, i.e., $\| P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \| \geq n$ Therefore, modular diagnosability focuses only on traces where events from module G_i, which is the module where the fault originates, occur with some regularity. Consequently, the notion of modular diagnosability is weaker than the notion of monolithic diagnosability since more languages will satisfy modular than monolithic diagnosability.

Our primary motivation for defining modular diagnosability is to ensure that after a fault occurs in one of the system modules, detection and isolation of that fault is only required along continuations that involve events from the given module. It is reminiscent of the familiar "persistency of excitation" condition in system identification. In other words, continuations that entirely exclude the module where the fault originates from cannot lead to a violation of modular diagnosability. (Recall that the approach that we propose assumes that faults do not bring the system, or any of its modules, to a halt.)

For the sake of illustration, let us consider a simple Local Area Network (LAN) composed of several interconnected computers. The LAN is the system to be diagnosed and the computers attached to it represent the local systems or modules. The faults or special events to be detected are "illegal" intrusions into the LAN. Therefore if an (unobservable) intrusion occurs at one of the computers and that computer does not exhibit any behavior after the intrusion, i.e., the local site does not supply any observable events, then clearly this intrusion does not need to be diagnosed since it is not exploited by the intruder. On the other hand, if the intruder takes advantage of its trespass, then it is essential to diagnose the intrusion. In other words, we expect that the intruder will sufficiently exert the afflicted computer so that the intrusion in the LAN can eventually be detected. This concept of "sufficient exertion" is similar to the one used in signature-based Intrusion

[3] Other researchers have also independently suggested this change (Yoo and Garcia, 2004).

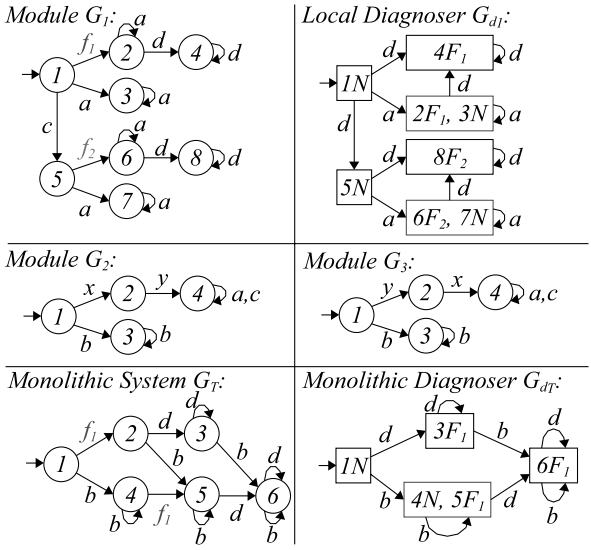


Fig. 1 Modular vs. monolithic diagnosability example

Detection Systems (IDS) where the signatures are specific sequences of (observable) events, cf. (Coolen and Luiijf, 2002). IDS gather sequences of observable events and verify if they match any of the sequences in IDS signature databases. In order to potentially match a signature, IDS require arbitrarily long exertion of the targeted local system.

The following example illustrates the difference between modular and monolithic diagnosability.

Example 1: Let $T = \{1, 2, 3\}$. Consider the system modules G_1, G_2, and G_3, the monolithic system $G_T = G_1 \parallel G_2 \parallel G_3$, the monolithic diagnoser G_{d_T}, the local diagnoser G_{d_1}, and their respective event sets Σ_1, Σ_2, Σ_3, Σ_T, Σ_{d_T}, and Σ_{d_1}. These models are depicted in Fig 1. We have $\Sigma_{uo} = \{f_1, f_2\}$, $\Sigma_o = \{a, b, c, d, x, y\}$, $\Sigma_1 = \{a, c, d, f_1, f_2\}$, $\Sigma_2 = \Sigma_3 = \{a, b, c, x, y\}$, $\Sigma_T = \{b, d, f_1\}$, $\Sigma_{d_T} = \{b, d\}$, and $\Sigma_{d_1} = \{a, c, d\}$ The diagnoser G_{d_1} contains cycles of F_1- and F_2-uncertain states, where F_1 and F_2 are the label associated with the fault events f_1 and $f_2 \in \Sigma_1$, respectively. They are identified as F_1- and F_2-indeterminate cycles, respectively. Therefore the diagnoser G_{d_1} is not (monolithically/modularly) diagnosable *w.r.t. Σ_{o_1} and Σ_{f_1}. We now investigate if the complete system is monolithically or modularly diagnosable. The diagnoser G_{d_T} contains a cycle of F_1-uncertain states. We check the necessary and sufficient conditions of modular and monolithic diagnosability. The diagnoser G_{d_T} contains a cycle formed by the self-loop $b \in \Sigma_2 \cap \Sigma_3$ at the F_1-uncertain state $q = \{4N, 5F_1\}$. It can be verified that this is an F_1-indeterminate cycle in G_{d_T}. Therefore the system G_T is* not monolithically diagnosable *w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. On the other hand, there does not exist an F^{M_1}-indeterminate cycle in G_{d_T} since $f_1 \in \Sigma_1$ and $b \notin \Sigma_1$. Hence G_T is* modularly diagnosable *w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. Intuitively, the above results are clear since the cycle of uncertain states in the (monolithic) diagnoser*

G_{d_T} is only composed of events from subsystems G_2 and G_3 while the fault to be diagnosed originates from module G_1.

We formalize the relationship between modular and monolithic diagnosability in the following theorem.

THEOREM 2

Part 1: Let $T = \{1, \ldots, I\}$, $S \subseteq T$, and $G_S = \|_{z\in S} G_z$. If the language $\mathcal{L}(G_S)$ is monolithically diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$ then $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$.
Part 2: Let $T = \{1, \ldots, I\}$, $S \subseteq T$, $G_S = \|_{z\in S} G_z$, and $i \in S$. If the language $\mathcal{L}(G_i)$ is monolithically diagnosable w.r.t. Σ_{o_i} and Σ_{f_i} then $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}.

Proof: Theorem 2 Part 1: We prove the contrapositive, i.e., if $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$, then $\mathcal{L}(G_S)$ is not monolithically diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$. $\mathcal{L}(G_S)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$ implies that $\exists i \in S$, $\exists f \in \Sigma_{f_i}$, $\exists s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\forall n \in \mathbb{N}$, $\exists t \in \mathcal{L}(G_S)/s$ such that $\| P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \| \geq n \Rightarrow D(st) = 0$. Since $\| P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \| \geq n$ implies $\| P_{\{\Sigma_S,\Sigma_{o_S}\}}(t) \| \geq n$, then $\mathcal{L}(G_S)$ is not monolithically diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$.

■

Proof: Theorem 2 Part 2: By Part 1 of Theorem 2, if the language $\mathcal{L}(G_i)$ is monolithically diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, then $\mathcal{L}(G_i)$ is modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}. We now prove the following: if $\mathcal{L}(G_i)$ is modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i} then $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}. We prove the contrapositive of the above statement, i.e., if $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, then $\mathcal{L}(G_i)$ is not modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}.

$\mathcal{L}(G_S)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i} implies that $\exists f \in \Sigma_{f_i}$, $i \in S$, $\exists s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\forall n \in \mathbb{N}$, $\exists t \in \mathcal{L}(G_S)/s$ such that $\| P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \| \geq n \Rightarrow D(st) = 0$, i.e., $\exists \omega_1, \omega_2 \in \mathcal{L}(G_S)$ such that

- $f \in \omega_1$ where $f \in \Sigma_{f_i}$, $i \in S$, $\omega_1 = s_1 t_1$, and s_1 ends with f,
- $f \notin \omega_2$,
- $P_{\{\Sigma_S,\Sigma_{o_S}\}}(\omega_1) = P_{\{\Sigma_S,\Sigma_{o_S}\}}(\omega_2)$, and
- $P_{\{\Sigma_S,\Sigma_{o_i}\}}(t_1)$ is arbitrarily long.

Let $\omega_1^i = P_{\{\Sigma_S,\Sigma_i\}}(\omega_1)$, $s_1^i = P_{\{\Sigma_S,\Sigma_i\}}(s_1)$, $t_1^i = P_{\{\Sigma_S,\Sigma_i\}}(t_1)$, and $\omega_2^i = P_{\{\Sigma_S,\Sigma_i\}}(\omega_2)$. Hence we have the following:

- $\omega_1^i, \omega_2^i \in \mathcal{L}(G_i)$,
- $f \in \omega_1^i$ where $\omega_1^i = s_1^i t_1^i$ and s_1^i ends with f,
- $f \notin \omega_2^i$,
- $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_1^i) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_2^i)$, and
- $P_{\{\Sigma_i,\Sigma_{o_i}\}}(t_1^i)$ is arbitrarily long.

Therefore $\mathcal{L}(G_i)$ is not modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}. ■

Preliminary Discussion

In the previous section we presented and motivated the notion of modular diagnosability and the conditions necessary and sufficient to ensure it. The verification process, based on the above results, requires the construction of the monolithic system (i.e., the parallel composition of the modules) and its corresponding diagnoser. In the case of large and complex systems both constructions can be computationally demanding. Therefore, we are interested in verifying the modular diagnosability property without necessarily building the monolithic system and its diagnoser. This section gives the intuition and the steps of the thinking process towards this goal. The outcome is a set of properties and an algorithm presented in the Properties of Modular Diagnosability section and the Test for Modular Diagnosability section, respectively.

Our objective is to determine whether the monolithic system is modularly diagnosable without constructing it and checking its diagnoser, if possible. Therefore, the first task is to determine (i) the origin of the set of traces that form an indeterminate cycle in the monolithic system, i.e., the modules responsible for the formation of the indeterminate cycle, and (ii) the survival of such traces when several of the modules or all of them are considered in the parallel composition. We call "*troublesome traces*," the traces forming an indeterminate cycle in a local diagnoser. In the sequel (the Properties of Modular Diagnosability section), we establish the following results: (a) if an indeterminate cycle exists in the monolithic diagnoser then necessarily one of the local diagnosers contains an indeterminate cycle; (b) if none of the local diagnosers contains an indeterminate cycle then the monolithic diagnoser does not contain an indeterminate cycle; and (c) if an indeterminate cycle exists in a local diagnoser then it may or may not exist in the monolithic diagnoser. The above results cover all possible outcomes (concerning indeterminate cycles in local diagnosers) that can occur when a system's modules are composed in parallel. Therefore, we focus on Case (c).

Case (c) yields the following objective. Starting from an individual module whose diagnoser contains an indeterminate cycle, our goal is to determine if this indeterminate cycle (call it ICX) is going to survive in the monolithic diagnoser without constructing it, if possible. In other words, we have to determine the reachability of the troublesome traces formed by ICX when the system's modules are composed in parallel. Since traces in a parallel composition operation are synchronized via common events, traces can be blocked only by common events. Thus, it suffices to consider in a module only the traces corresponding to an indeterminate cycle and only the common events in these traces. Then, the first verification step is to create a parallel composition involving (i) the module under consideration (where we keep only the common events of ICX), and (ii) all modules that have events in common with the ones of ICX. Two outcomes are possible: (1) the indeterminate cycle ICX is not present (i.e., not reachable) in the resulting parallel composition; in this case we say that ICX is blocked by the parallel composition process; and (2) the indeterminate cycle ICX is present (i.e., reachable) in the resulting parallel composition; in this case we need to further verify whether or not ICX is blocked indirectly via other traces by incrementally integrating more modules. Thus, in the case of outcome (2), the next verification steps are as follows. At each increment we add the modules

that have an event in common with the events in the previously resulting automaton, do the parallel composition, and check if the indeterminate cycle exists due to ICX in the new resulting automaton. We need to repeat these incremental steps until one of the following outcomes occur: (i) the indeterminate cycle due to ICX is blocked; (ii) the rest of the modules and the automaton resulting from the previous parallel composition do not have any events in common; or (iii) all the modules have been accounted for. In the Test for Modular Diagnosability section we prove that this approach achieves the above-stated goal.

Properties of Modular Diagnosability

This section presents a set of preliminary results that are essential in establishing the proof of Theorem 4, which asserts that MDA presented in this paper determines correctly whether or not $\mathcal{L}(G_T)$ is modularly diagnosable. We define $ND = T \setminus D$ where $D = \{z : \mathcal{L}(G_z)$ is monolithically diagnosable w.r.t. Σ_{o_z} and $\Sigma_{f_z}\}$.

LEMMA 1 *If* $\forall i \in S \cap ND$, $\mathcal{L}(G_S)$ *is modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in S)$ *and* Σ_{f_i} *then* $\mathcal{L}(G_S)$ *is modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in S)$ *and* $(\Sigma_{f_z} : z \in S)$.

Proof: We assume that $\forall i \in S \cap ND$, $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}. Therefore Definition 1 is satisfied $\forall i \in S \cap ND$ and $\forall f \in \Sigma_{f_i}$. By Part 2 of Theorem 2, Definition 1 is satisfied for $\forall i \in S \cap D$ and $\forall f \in \Sigma_{f_i}$. Thus $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$. ■

LEMMA 2 *If* $\forall i \in T \cap ND$, $\exists S \subseteq T$ *s.t.* $i \in S$ *and* $\mathcal{L}(G_S)$ *is modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in S)$ *and* Σ_{f_i} *then* $\mathcal{L}(G_T)$ *is modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in T)$ *and* $(\Sigma_{f_z} : z \in T)$.

Proof: We prove the contrapositive: if $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$ then $\exists i \in T \cap ND$ s.t. $\forall S \subseteq T$ with $i \in S$, $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}.

From Definition 1 and Part 2 of Theorem 2, $\mathcal{L}(G_T)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$ implies that $\exists i \in T \cap ND$, $\exists s, s' \in \mathcal{L}(G_T)$, $\exists f \in \Sigma_{f_i}$ s.t. $f \in s$ and $f \notin s'$, $P_{\{\Sigma_T, \Sigma_{o_T}\}}(s) = P_{\{\Sigma_T, \Sigma_{o_T}\}}(s')$, and $P_{\{\Sigma_T, \Sigma_{o_i}\}}(t)$ is arbitrarily long. Also, $\forall S \subseteq T$ s.t. $i \in S$ we have the following: $P_{\{\Sigma_T, \Sigma_{o_S}\}}(s) = \mathrm{P}_{\{\Sigma_T, \Sigma_{o_S}\}}(s')$ and $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s) = P_{\{\Sigma_T, \Sigma_{o_i}\}}(s')$ since $P_{\{\Sigma_T, \Sigma_{o_T}\}}(s) = P_{\{\Sigma_T, \Sigma_{o_T}\}}(s')$. Furthermore, $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s')$ is arbitrarily long since $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s)$ is arbitrarily long. Let $s_x, s'_x \in \mathcal{L}(G_S)$ s.t. $s_x = P_{\{\Sigma_T, \Sigma_S\}}(s)$ and $s'_x = P_{\{\Sigma_T, \Sigma_S\}}(s')$. Then $f \in s_x$ and $f \notin s'_x$. Also $P_{\{\Sigma_S, \Sigma_{o_i}\}}(s_x)$, $P_{\{\Sigma_S, \Sigma_{o_i}\}}(s'_x)$ are arbitrarily long since $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s)$, $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s')$ are arbitrarily long.

In summary, $\exists i \in T \cap ND$ s.t. $\forall S \subseteq T$ with $i \in S$, $\exists s_x, s'_x \in \mathcal{L}(G_S), \exists f \in \Sigma_{f_i}, f \in s_x, f \notin s'_x$, $P_{\{\Sigma_S, \Sigma_{o_S}\}}(s_x) = P_{\{\Sigma_S, \Sigma_{o_S}\}}(s'_x)$, and $P_{\{\Sigma_S, \Sigma_{o_i}\}}(s_x)$ is arbitrarily long. Therefore $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}. ■

COROLLARY 1 *If* $\forall i \in T$, $\mathcal{L}(G_i)$ *is monolithically diagnosable w.r.t.* Σ_{o_i} *and* Σ_{f_i}, *then* $\mathcal{L}(G_T)$ *is modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in T)$ *and* $(\Sigma_{f_z} : z \in T)$.

Proof: Corollary 1 is a particular case of Lemma 2 when $S = \{i\}$. ■

An elementary F_m-indeterminate cycle, $m \in \{1,\ldots,M\}$, is formed by (i) a sequence of F_m-uncertain states and (ii) possibly several sequences of events that form the cycle and satisfy the F_m-indeterminate cycle definition. We call EIC_z, $z \in \{1,\ldots,Z\}$, such cycles and t_y^z, $y \in \{1,\ldots,Y_z\}$, their corresponding sequences of events, where Z represents the total number of elementary F_m-indeterminate cycles in the diagnoser G_{d_i}, $i \in ND$, and Y_z represents the total number of sequences of events that satisfy the indeterminate cycle definition for the particular EIC_z.

For each $i \in ND$, we number and name $SEQ_1, SEQ_2, \ldots, SEQ_{X_i}$ all sequences of events t_y^z, $y = 1,\ldots,Y_z$ and $z = 1,\ldots,Z$. Therefore, each SEQ_x, $x \in \{1,\ldots,X_i\}$, is associated with: (i) one F_m-indeterminate cycle; (ii) one fault of type m, $m \in \{1,\ldots,M\}$; (iii) one corresponding sequence of states $Q^x = q_1 \ldots q_{N_x}$; and (iv) one sequence of events t_y^z, $z \in \{1,\ldots,Z\}$, $y \in \{1,\ldots,Y_z\}$, that form the cycle. We attach the label M_x, $x \in \{1,\ldots,X_i\}$, $i \in ND$, to states $q \in Q^x$ in G_{d_i}.

The following lemma is a specialized form of Lemma 2.

LEMMA 3 *Consider $S \subseteq T$, SEQ_x, $x \in \{1,\ldots,X_i\}$, $i \in S \cap ND$, and any two arbitrarily long traces $\omega_x, \omega_x' \in \mathcal{L}(G_i)$ such that: (i) ω_x, ω_x' lead to the indeterminate cycle associated with SEQ_x in G_{d_i}; (ii) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x') = ss_1s_2s_1 = sSEQ_xs_1$ where $SEQ_x = s_1s_2$; (iii) $f_m \in \omega_x$, $f_m \notin \omega_x'$, and f_m corresponds to the fault type associated with SEQ_x. If $\nexists\, \omega_S, \omega_S' \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S') = \omega_x'$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long, then $\nexists \omega, \omega' \in L(G_T)$ such that $P_{\{\Sigma_T,\Sigma_i\}}(\omega) = \omega_x$, $P_{\{\Sigma_T,\Sigma_i\}}(\omega') = \omega_x'$, and $P_{\{\Sigma_T,\Sigma_{o_i}\}}(\omega)$ is arbitrarily long.*

Proof: We prove by contradiction. By assumption, $\exists S \subseteq T$, $\exists SEQ_x$, $x \in \{1,\ldots,X_i\}$, $i \in S \cap ND$, and there exist two arbitrarily long traces $\omega_x, \omega_x' \in \mathcal{L}(G_i)$ such that: (i) ω_x, ω_x' lead to the indeterminate cycle associated with SEQ_x in G_{d_i}; (ii) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x') = sSEQ_xs_1$ where $SEQ_x = s_1s_2$; (iii) $f_m \in \omega_x$, $f_m \notin \omega_x'$, and f_m corresponds to the fault type associated with SEQ_x. Suppose that (iv) $\omega_S, \omega_S' \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S') = \omega_x'$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long and (v) $\exists \omega$, $\omega' \in L(G_T)$ such that $P_{\{\Sigma_T,\Sigma_i\}}(\omega) = \omega_x, P_{\{\Sigma_T,\Sigma_i\}}(\omega') = \omega_x'$, and $P_{\{\Sigma_T,\Sigma_{o_i}\}}(\omega)$ is arbitrarily long.

By assumption (v) and the natural projection definition, $\exists \omega_S, \omega_S' \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_T,\Sigma_S\}}(\omega) = \omega_S$, $P_{\{\Sigma_T,\Sigma_S\}}(\omega') = \omega_S'$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long. Furthermore we have $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = P_{\{\Sigma_S,\Sigma_i\}}[P_{\{\Sigma_T,\Sigma_S\}}(\omega)] = P_{\{\Sigma_T,\Sigma_i\}}(\omega) = \omega_x$ and $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S') = \omega_x'$. Therefore $\exists \omega_S, \omega_S' \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S') = \omega_x'$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long, which yields the desired contradiction. ■

Remark 3: When the hypothesis of Lemma 3 holds, we say that the indeterminate cycle associated with SEQ_x is "Not Reachable" in G_S and G_T. In other words, the coupling of module G_i with the remainder of the system results in the elimination of the traces in $\mathcal{L}(G_i)$ that lead to that indeterminate cycle.

COROLLARY 2 *If the hypothesis of Lemma 3 holds for all x, $x \in \{1,..,X_i\}$, then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}.*

LEMMA 4 *If* $\exists i \in T$, $\exists S \subseteq T$ *s.t.* $i \in S$, $S^c = T \setminus S$, *and* $\exists \omega_S, \omega'_S \in L(G_S)$ *such that:*

(*i*) ω_S, ω'_S *violate the modular diagnosability of* $\mathcal{L}(G_S)$ *w.r.t.* $(\Sigma_{o_z} : z \in S)$ *and* Σ_{f_i}, *and* $P_{\{\Sigma_S, \Sigma_{o_i}\}}(\omega_S) = sSEQ_x s_1$, $SEQ_x = s_1 s_2$, $x \in \{1, ..., X_i\}$;

(*ii*) $\forall \sigma_x \in \omega_S$, $\forall \sigma_y \in S^c$, $\sigma_x \neq \sigma_y$; *then* $\mathcal{L}(G_T)$ *is not modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.

Proof: We have $G_T = G_S \parallel G_{S^c}$. Build $G_{\tilde{S}}$ s.t. $\mathcal{L}(G_{\tilde{S}}) := \overline{\{\omega_S, \omega'_S\}}$. Define $G_{\tilde{T}} := G_{\tilde{S}} \parallel G_{S^c}$. By the definition of $G_{\tilde{T}}$ and assumption (ii), $\exists \omega, \omega' \in \mathcal{L}(G_{\tilde{T}})$ s.t.

- $P_{\{\Sigma_{\tilde{T}}, \Sigma_{\tilde{T}}\}}(\omega) = \omega_S$, $P_{\{\Sigma_{\tilde{T}}, \Sigma_{\tilde{T}}\}}(\omega') = \omega'_S$,
- $P_{\{\Sigma_{\tilde{T}}, \Sigma_{o_{\tilde{T}}}\}}(\omega) = P_{\{\Sigma_{\tilde{T}}, \Sigma_{o_{\tilde{T}}}\}}(\omega')$,
- $f \in \omega$, $f \notin \omega'$, where $f \in \Sigma_{f_i}$, and
- $P_{\{\Sigma_{\tilde{S}}, \Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long because $P_{\{\Sigma_S, \Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long as it violates modular diagnosability by assumption (i).

Since $\mathcal{L}(G_{\tilde{S}}) \subseteq \mathcal{L}(G_T), \omega, \omega' \in \mathcal{L}(G_{\tilde{T}})$ implies $\omega, \omega' \in \mathcal{L}(G_T)$ and therefore ω, ω' violate the modular diagnosability of $\mathcal{L}(G_T)$ w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. ■

We make the following observation regarding the proof of Lemma 4. Any $\omega \in \mathcal{L}(G_{\tilde{T}})$ is built from $\omega_S \in \mathcal{L}(G_{\tilde{S}})$ by interleaving events from Σ_{S^c} according to the transition structure of G_{S^c}. Hence, since $\Sigma_{\tilde{S}} \cap \Sigma_{S^c} = \emptyset$, we can build ω' from ω'_S by doing the same interleaving as when building ω from ω_S. The resulting ω' necessarily satisfies $P_{\{\Sigma_{\tilde{T}}, \Sigma_{o_{\tilde{T}}}\}}(\omega) = P_{\{\Sigma_{\tilde{S}}, \Sigma_{o_{\tilde{T}}}\}}(\omega')$.

Remark 4: When the hypothesis of Lemma 4 holds, we say that the indeterminate cycle associated with SEQ_x is "Reachable" in G_S and G_T. In other words, the coupling of module G_i with the remainder of the system results in the propagation of the traces in $\mathcal{L}(G_i)$ that lead to that indeterminate cycle.

LEMMA 5 *Assume* $\exists i \in T$, $\exists S \subseteq T$ *s.t.* $i \in S$, $S^c = T \setminus S$, *and* $\exists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ *such that:*

(*i*) ω_S, ω'_S *violate the modular diagnosability of* $\mathcal{L}(G_S)$ w.r.t. $(\Sigma_{o_z} : z \in S)$ *and* f_i, *and* $P_{\{\Sigma_S, \Sigma_{o_i}\}}(\omega_S) = sSEQ_x s_1$, $SEQ_x = s_1 s_2$, $x \in \{1, \ldots, X_i\}$;

(*ii*) $\exists \sigma_x \in \omega_S$ *and* $\exists \sigma_y \in \Sigma_{S^c}$ *such that* $\sigma_x = \sigma_y$. *From hypotheses (i) and (ii), we cannot conclude whether* $\mathcal{L}(G_T)$ *is or is not modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in T)$ *and* Σ_{f_i}.

Proof: We have $G_T = G_S \parallel G_{S^c}$. By assumption (ii), two (exhaustive) cases are possible. Define $\omega, \omega' \in \Sigma_T^*$ such that:

- $P_{\{\Sigma_T, \Sigma_S\}}(\omega) = \omega_S$, $P_{\{\Sigma_T, \Sigma_S\}}(\omega') = \omega'_S$,
- $P_{\{\Sigma_T, \Sigma_{o_T}\}}(\omega) = P_{\{\Sigma_T, \Sigma_{o_T}\}}(\omega')$,
- $f \in \omega$, $f \notin \omega'$, where $f \in f_i$, and
- $P_{\{\Sigma_S, \Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long.

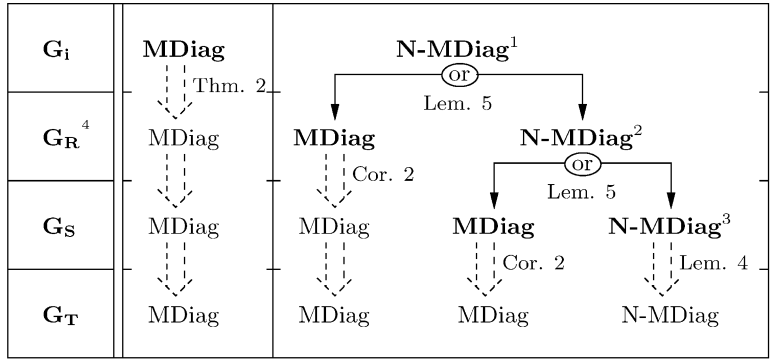
Case 1: If such ω, ω' exist in $\mathcal{L}(G_T)$, then $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}.

Case 2: On the other hand, if no such ω, ω' exist then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{fi}. ■

We discuss the intuition behind Lemma 5. Any $\omega \in \mathcal{L}(G_T)$ is built from $\omega_S \in \mathcal{L}(G_S)$ by interleaving events from Σ_{S^c} according to the transition structure of G_{S^c}. Hence, since $\Sigma_S \cap \Sigma_{S^c} \neq \emptyset$, any event $\sigma_x \in \Sigma_S \cap \Sigma_{S^c}$ may or may not be synchronized during the parallel composition $G_T = G_S \parallel G_{S^c}$. The existence of the traces ω, ω' in the proof of Lemma 5 depends on the outcome of such synchronization. The reason for stating as a lemma the fact that hypotheses (i) and (ii) are not conclusive regarding modular diagnosability is because this fact will be used in the logic and proof of MDA presented in the following section.

Remark 5: If $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i} then G_S necessarily satisfies the hypotheses of Lemmata 4 or 5.

In Fig 2, we depict the implications of Theorem 2, Corollary 2, and Lemmata 4, 5. The figure shows that if $\mathcal{L}(G_i)$ is modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, or $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}. If $\mathcal{L}(G_i)$ is not modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, or $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, then the output on the modular diagnosability of $\mathcal{L}(G_T)$ w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i} is uncertain unless G_S satisfies Lemma 4.



Legend:
MDiag : Modularly Diagnosable w.r.t. $(\Sigma_{o_z} : z \in X)$ and Σ_{f_i}, where G_X is the considered system.
N-MDiag: Not Modularly Diagnosable w.r.t. $(\Sigma_{o_z} : z \in X)$ and Σ_{f_i}.
: Two Possible Outputs (need to construct the pointed module).
: Direct Implication (no computation or construction needed).
Cor. : Corollary.
Lem. : Lemma.
Thm. : Theorem.

Notes:
1: Subsystem G_i necessarily satisfies hypothesis (ii) of Lemma 5.
2: We assume that G_R satisfies hypothesis (ii) of Lemma 5.
3: We assume that G_S satisfies hypothesis (ii) of Lemma 4.
4: $R \subset S \subseteq T$, $i \in R$.

Fig. 2 Properties of modular diagnosability.

Test for Modular Diagnosability

In the Modular Diagnosability section we presented the notion of modular diagnosability for modular discrete event systems and conditions necessary and sufficient to guarantee it. In this section, we propose a novel approach that tests modular diagnosability by incorporating incrementally, in a systematic manner, subsystems into the test. We prove that our approach provides the correct answer to the question "Is $\mathcal{L}(G_T)$ modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$?" in a finite number of steps. We proceed as follows. In the Modular Diagnosability Algorithm subsection we present the algorithm; in the Properties of MDA subsection we state and prove its properties; in the Discussion subsection and the Online Diagnosis section we present a discussion of the key steps of the algorithm and online diagnosis, respectively.

Modular Diagnosability Algorithm

We present a detailed statement of our Modular Diagnosability Algorithm (MDA). For the sake of clarity, MDA is broken into three algorithms. Algorithm 1 is the core of MDA; it calls Algorithm 2 to perform preliminary steps involving indeterminate cycles that could lead to a violation of modular diagnosability. The goal of Algorithm 2 is to identify all the indeterminate cycles that are present in the modules and yield a list of sequences of states and events that is used in the other algorithms. Algorithm 1 also calls Algorithm 3 where the incremental analysis of each indeterminate cycle is performed.

Algorithm 1—MDA

1) *Let* $T = \{1, \ldots, I\}$. *Construct the local diagnosers* G_{d_i}, $i \in T$, *and search for indeterminate cycles. If,* $\forall i \in T$, $\mathcal{L}(G_i)$ *is monolithically diagnosable w.r.t.* Σ_{o_i} *and* Σ_{f_i}, *i.e., none of the local diagnosers* G_{d_i} *have F-indeterminate cycles, then stop and declare* $\mathcal{L}(G_T)$ *modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in T)$ *and* $(\Sigma_{f_z} : z \in T)$. *Else, go to Step 2.*
2) *Let* $ND = T \backslash D$, *where* $D = \{z : \mathcal{L}(G_z)\ \Sigma o_z$ and $\Sigma_{f_z}\}$ *is monolithically diagnosable w.r.t.* Σ_{o_z} *and* Σ_{f_z}. *Call* ***Preliminary Function*** *with argument {ND} and store its output. For each local diagnoser* G_{d_i}, $i \in ND$, *and for each sequence of traces* SEQ_x, $x \in \{1, \ldots, X_i\}$, *perform the Steps 2-a to 2-d:*
   - 2-a) *Mark with the label* M_x, $x \in \{1, \ldots, X_i\}$, $i \in ND$, *the states* $q \in Q^x$ *in* G_{d_i}. *The label* M_x *stands for "State of* G_{d_i} *part of the indeterminate cycle associated with* SEQ_x."
   - 2-b) *Construct*

$$G_{CM_i} = Obs(G_{d_i}, \Sigma_{CM_i}). \tag{10}$$

   A state of G_{CM_i} *is marked with label* M_x *if one or more of its state components are marked with* M_x.
   - 2-c) *Construct*

$$G_{ICM_x} = CoAc(G_{CM_i}, M_x). \tag{11}$$

The resulting event set of automaton G_{ICM_x} is denoted by Σ_{ICM_x}. Enlarge the set of events Σ_{ICM_x} by adding $\Sigma_{CM_i} \setminus \Sigma_{ICM_x}$ to it. The automaton G_{ICM_x} and its newly associated event set Σ_{CM_i} are hereafter represented by the notation $(G_{ICM_x}, \Sigma_{CM_i})$.

2-d) *Call* ***Reachability Function*** *with argument $\{i, SEQ_x, G_{ICM_x}, \Sigma_{ICM_x}, \Sigma_{CM_i}\}$. If the Reachability Function returns "Reachable" then stop and declare $\mathcal{L}(G_T)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. If the Reachability Function returns "Non-Reachable" then continue.*

3) *Stop and declare $\mathcal{L}(G_T)$ modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.*

ALGORITHM 2—Preliminary function {ND}

I) For each $i \in ND$, do the following:

i) Call EIC_z, $z \in \{1,\ldots,Z\}$, the elementary[4] indeterminate cycles in G_{d_i} and t^z_y, $y \in \{1,\ldots,Y_z\}$, their corresponding sequences of events, where Z represents the total number of elementary indeterminate cycles in diagnoser G_{d_i}, $i \in ND$, and Y_z represents the total number of sequences of events that satisfy the indeterminate cycle definition for the particular EIC_z.

ii) $\forall z \in \{1,\ldots,Z\}$, if $\exists y_1, y_2 \in \{1,\ldots,Y_z\}$ and t', t'' such that $y_1 < y_2$, $t^z_{y_1} = t't''$, and $t^z_{y_2} = t''t'$ then delete $t^z_{y_2}$ from the list as follows: let $n = y_2$ and

ii-a) $t^z_{y_n} := t^z_{y_{n+1}}$

ii-b) if $n + 1 = Y_z$ then delete t^z_{n+1}; otherwise let $n := n + 1$ and go to (ii-a).

iii) Number and name $SEQ_1, SEQ_2, \ldots, SEQ_{X_i}$ all sequences of events t^z_y, $y = 1, \ldots, Y_z$ and $z = 1,\ldots,Z$. To each SEQ_x, $x \in \{1,\ldots,X_i\}$, associate its corresponding F_m-indeterminate cycle, $m \in \{1,\ldots,M\}$, its corresponding sequence of states $Q^x = q_1 \ldots q_{N_x}$, and its corresponding sequence of events t^z_y, $z \in \{1,\ldots,Z\}$, $y \in \{1,\ldots,Y_z\}$, that form the cycle.

iv) If $\exists x'$, $x'' \in \{1,\ldots,X_i\}$, where $x' < x''$, and Q', Q'', SEQ', SEQ'' such that

- *$Q_{x'} = Q'\,Q''$ and $Q''_x = Q''\,Q'$,*
- *$SEQ_{x'} = SEQ'SEQ''$ and $SEQ_{x''} = SEQ''SEQ'$, and*
- *$|SEQ'| = |\,Q'|$,*

then concatenate $SEQ_{x'}$ and $SEQ_{x''}$ by doing the following steps. Add to the information associated with $SEQ_{x'}$ the information relative to $SEQ_{x''}$ [i.e., the F_m-indeterminate cycle and elementary cycle EIC_z of $SEQ_{x''}$, and the sequence of states $Q_{x''}$]. Similarly to step (ii), we delete $SEQ_{x''}$ from the list as follows: let $n = x''$ and

iv-a) $SEQ_n := SEQ_{n+1}$

iv-b) if $n + 1 = X_i$ then delete SEQ_{n+1}; otherwise let $n := n + 1$ and go to (iv-a).

II) Return to MDA with Preliminary Function {ND} = $\{SEQ_x, Q^x: x \in \{1,\ldots,X_i\}$ and $i \in ND\}$.

[4] A cycle is called elementary if no state appears more than once in it.

ALGORITHM 3—Reachability function $\{i, SEQ_x, G_{ICM_x}, \Sigma_{ICM_x}, \Sigma_{CM_i}\}$

A) *Let* $c := 1$, $B_c^x = \{i\}$, $S_c = B_c^x$, *and*

$$\widetilde{s}^x = P_{\{\Sigma_{o_i}, \Sigma_{CM_{o_i}}\}}(SEQ_x). \tag{12}$$

Let $c := c + 1$,

$$B_c^x = \{l : [\Sigma_{CM_l} \cap \Sigma_{ICM_x} \neq \emptyset \vee (l \in B_{c-1}^x)], l \in T\}, \tag{13}$$

and

$$S_c = B_c^x \setminus B_{c-1}^x. \tag{14}$$

B) *Construct*

$$G_{mod_c^x} = (G_{ICM_x}, \Sigma_{CM_i}) \parallel (\|_{l \in B_c^x, l \neq i} G_{CMl}). \tag{15}$$

If there does not exist in $G_{mod_c^x}$ *a cycle of states labeled* M_x *then return to MDA with Reachability Function* $\{i, SEQ_x, G_{ICM_x}, \Sigma_{ICM_x}, \Sigma_{CM_i}\} = \{Non\text{-}\ Reachable\}$; *otherwise denote by* $s_1^c, s_2^c, \ldots, s_P^c$ *the sequences of events that describe such elementary cycles and go to step C.*

C) $\forall p \in \{1, \ldots, P\}$, *let*

$$\widetilde{s}_p^c = P_{\{\Sigma_{CM_{o_S}}, \Sigma_{CM_{o_i}}\}}(s_p^c), S := B_c^x. \tag{16}$$

If $\exists p \in \{1, \ldots, P\}$ *such that* $\widetilde{s}^x = \widetilde{s}_p^c$ *or if* $\exists s'^x$, s''^x, *and* $p \in \{1, \ldots, P\}$ *such that* $\widetilde{s}^x = s'^x s''^x$ *and* $s'^x s''^x = \widetilde{s}_p^c$, *then go to Step D; otherwise return to MDA with Reachability Function* $\{i, SEQ_x, G_{ICM_x}, \Sigma_{ICM_x}, \Sigma_{CM_i}\} = \{Non\text{-}Reachable\}$.

D) *Construct*

$$\widetilde{G}_c = CoAc(G_{mod_c^x}, M_x). \tag{17}$$

E) *Let* $\widetilde{\Sigma}_c$ *be the event set of* $\widetilde{G}_c$.
Let $c := c + 1$ *and*

$$B_c^x = \{l : [(\Sigma_{CM_l} \cap \widetilde{\Sigma}_{c-1} \neq \emptyset) \vee (l \in B_{c-1}^x)], l \in T\}. \tag{18}$$

Define

$$S_c = B_c^x \setminus B_{c-1}^x. \tag{19}$$

If $S_c \neq \emptyset$ *then go to step B; otherwise declare the indeterminate cycle associated with* SEQ_x *"Reachable" and return to MDA with Reachability Function* $\{i, SEQ_x, G_{ICM_x}, \Sigma_{ICM_x}, \Sigma_{CM_i}\} = \{Reachable\}$.

A Simple Example

For the sake of completeness, we apply the algorithm to the system used in the straightforward example presented in the Modular Diagnosability section, cf. Example 1 and Fig. 1. We present a summary of the results (see Contant et al., (2004a) for a step-by-step explanation).

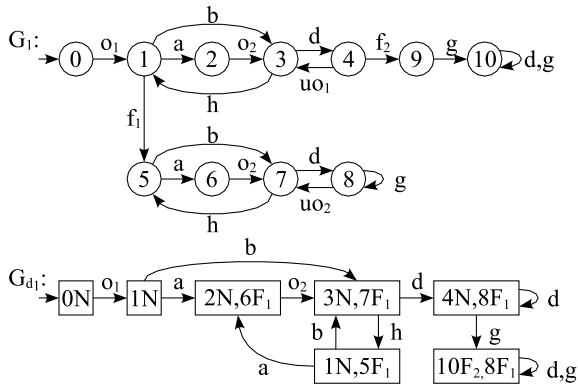


Fig. 3 Module G_1 and local diagnoser G_{d_1}

There exist two indeterminate cycles in the local diagnoser G_{d_1}, cf. Fig. 1 and, from Algorithm 2, two troublesome traces need to be checked. Therefore Algorithm 3 needs to be applied for each trace. The resulting machines $G_{mod_2^1}$ and $G_{mod_2^2}$ do not contain a cycle of states labeled M_1 and M_2, respectively. Therefore we declare the monolithic system, $\mathcal{L}(G_T)$, modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.

A Detailed Illustrative Example

For the sake of clarity, we provide an example that illustrates more thoroughly the steps of the algorithm. The considered system is composed of eight modules G_1 to G_8. For the sake of simplicity, the modules G_2 to G_8 contain only common events (and implicitly no fault events). Therefore $\forall i \geq 2$ we have $G_i \equiv G_{d_i} \equiv G_{CM_i}$. Modules G_1 and G_{CM_2} to G_{CM_8} are presented in Figs. 3 and 4.

- MDA Step (1): we construct the local diagnoser G_{d_1}, cf. Fig. 3. There exist several indeterminate cycles. Hence, we go to Step (2).
- MDA Step (2): we perform the Preliminary Function.
- Preliminary Function Step (I-i): we list the elementary indeterminate cycles (for all fault types) and the corresponding sequences of events, cf. Table 1. A cycle may consist of several sequences of events (e.g., EIC_4), and may have several entry points (e.g., EIC_1).

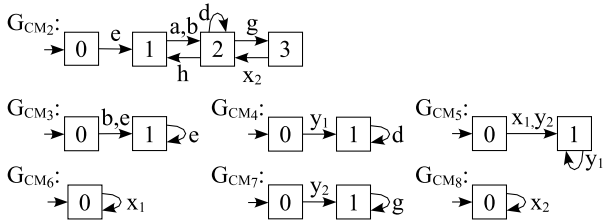


Fig. 4 Modules G_{CM_2} to G_{CM_8}

Table 1 Preliminary function, Step (I-i): elementary cycles

EIC_z	EIC_1	EIC_2	EIC_3
t_y^z	$t_1^1 = o_2ha$, $t_2^1 = hao_2$	$t_1^2 = hb$	$t_1^3 = d$
EIC_z	EIC_4	EIC_5	
t_y^z	$t_1^4 = d$, $t_2^4 = g$	$t_1^5 = d$, $t_2^5 = g$	

- Preliminary Function Step (I-ii): we delete all sequences of events that correspond to different entry points (viz., t_2^1 from EIC_1). In other words, we remove the sequences that are cyclic permutations of other sequences, cf. Table 3. For the sake of clarity we name the states in G_{d_1} as indicated in Table 2.
- Function Step (I-iii): we form a list of the troublesome traces, SEQ_x, cf. Table 3.
- Preliminary Function Step (I-iv): we cluster the sequences of events and states that are cycle permutations of one another. Then we form a list of troublesome traces, SEQ_x, cf. Table 4 and return to MDA Step (2).
- MDA Steps (2-a) to (2-d): we repeat these steps for each sequence of traces SEQ_x.

To avoid redundancy we present only the first of the five sequences. We refer the reader to Contant et al. (2004a) for a complete description of the example.

Sequence $SEQ_1 = o_2ha$

- MDA Step (2-a): we mark with the label M_1 the states Q^1 in G_{d_1}, cf. Fig. 5.
- MDA Step (2-b): we construct G_{CM_1} (see Fig. 6).
- MDA Steps (2-c): we construct G_{ICM_1} (see Fig. 7).
- MDA Step (2-d): we call the Reachability Function with $i = 1$, $x = 1$.
- Reachability Function Step (A): $B_1^1 = \{1\}$, $S_1 = B_1^1$, and $\tilde{s}^1 = P_{\{\Sigma_{o_1}, \Sigma_{CM_{o_1}}\}}(o_2ha) = ha$. Let $c := 2$, $B_2^1 = \{l : [\Sigma_{CM_1} \cap \{a, b, h\} \neq \emptyset \vee (l \in B_1^1)],\ l \in T\} = \{1, 2, 3\}$, and $S_2 = B_2^1 \setminus B_1^1 = \{2, 3\}$.
- Reachability Function Step (B): we construct $G_{mod_2^1} = (G_{ICM_1}, \Sigma_{CM_1}) \parallel G_{CM_2} \parallel G_{CM_3}$, cf. Fig. 8. There exists a cycle of states labeled M_1. We denote this cycle by $s_1^2 = ha$ and go to Step (C). [N.B.: $(G_{ICM_1}, _{CM_1})$ means that the set of events of G_{ICM_1}, $\{a, b, h\}$, is augmented with $\Sigma_{CM_1} = \{a, b, d, g, h\}$. This modification allows us to block event d and g from occurring in $G_{mod_2^1}$. As seen at the next iteration, only three modules instead of six will be considered. The reason to augment the set of events is to block directly the events that would be blocked if we were to compose the complete system.
- Reachability Function Step (C): $\tilde{s}_1^2 = P_{\{\Sigma_{CMo_S}, \Sigma_{CM_{o_1}}\}}(ha) = \tilde{s}^1 = ha$. Thus we go to Step (D).
- Reachability Function Step (D): we construct $\widetilde{G}_2 = CoAc(G_{mod_2^1}, M_1)$. We have $\widetilde{G}_2 \equiv G_{mod_2^1}$, cf. Fig. 8.

Table 2 States and state components of G_{d_1}

States	0	1	2	3	4	5	6
States Components	$0N$	$2N6F_1$	$3N7F_1$	$1N5F_1$	$4N8F_1$	$10F_28F_1$	$1N$

Table 3 Preliminary function, Step (I-ii and I-iii): sequence of events SEQ_x

SEQ_x	EIC_z	F_m-Indet. Cycle	Q^x
$SEQ_1 = t_1^1 = o_2ha$	EIC_1	F_1-Indet. Cycle	$Q^1 = 1, 2, 3$
$SEQ_2 = t_1^2 = hb$	EIC_2	F_1-Indet. Cycle	$Q^2 = 2, 3$
$SEQ_3 = t_1^3 = d$	EIC_3	F_1-Indet. Cycle	$Q^3 = 4$
$SEQ_4 = t_1^4 = d$	EIC_4	F_1-Indet. Cycle	$Q^4 = 5$
$SEQ_5 = t_2^4 = g$	EIC_4	F_1-Indet. Cycle	$Q^5 = 5$
$SEQ_6 = t_1^5 = d$	EIC_5	F_2-Indet. Cycle	$Q^6 = 5$
$SEQ_7 = t_2^5 = g$	EIC_5	F_2-Indet. Cycle	$Q^7 = 5$

- Reachability Function Step (E): let $c := 3$ and $B_3^1 = \{l : [(\Sigma_{CM_l} \cap \{a, e, h\} \neq \emptyset) \vee (l \in B_2^3)], l \in T\} = \{1, 2, 3\}$. $S_3 = B_3^1 \setminus B_2^1 = \emptyset$. Since $S_3 = \emptyset$ then we declare the indeterminate cycle associated with SEQ_1 "Reachable," return to MDA Step (2-d) and declare $\mathcal{L}(G_T)$, the complete system, not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.

Properties of MDA

THEOREM 3 *MDA Returns an answer in a finite number of steps.*

Proof: Since there is a finite number I of subsystems and B_c^x is monotonically increasing by equation (18), the loop in Steps (B)–(E) of the Reachability Function is carried out a finite number of times. Thus **Reachability Function** returns an answer in a finite number of steps for every sequence of events SEQ_x. Since there is a finite number of sequences of events SEQ_x, MDA returns an answer in a finite number of steps. ■

THEOREM 4 *MDA Returns the correct answer, namely, whether $\mathcal{L}(G_T)$ is or is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.*

Proof: We prove that Steps 1, 2-d, and 3 of MDA and steps B, C, and E of the Reachability Function return the correct answer. The correctness of Steps 2-d and 3 of MDA depends on the correctness of the Reachability Function. Thus, we proceed as follows. We first prove that the Reachability Function returns the correct answer to the question: "Is the indeterminate cycle associated with SEQ_x reachable in the global system behavior $\mathcal{L}(G_T)$?." Then we prove the correctness of Steps 1, 2-d, and 3 of MDA, using the correctness of the Reachability Function.

Table 4 Preliminary function, Step (I-iv), and MDA, Step (2-a)

SEQ_x	EIC_z	F_m-Indet. Cycle	Q^x	M_x
$SEQ_1 = t_1^1 = o_2ha$	EIC_1	F_1-Indet. Cycle	$Q^1 = 1, 2, 3$	M_1
$SEQ_2 = t_1^2 = hb$	EIC_2	F_1-Indet. Cycle	$Q^2 = 2, 3$	M_2
$SEQ_3 = t_1^3 = d$	EIC_3	F_1-Indet. Cycle	$Q^3 = 4$	M_3
$SEQ_4 = t_1^4 = t_1^5 = d$	EIC_4 & EIC_5	F_1- and F_2-Indet. Cycle	$Q^4 = 5$	M_4
$SEQ_5 = t_2^4 = t_2^5 = g$	EIC_4 & EIC_5	F_1- and F_2-Indet. Cycle	$Q^5 = 5$	M_5

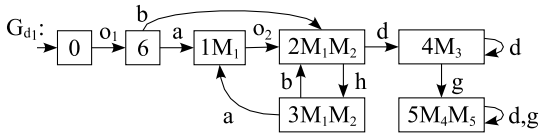


Fig. 5 Local diagnoser G_{d_1} with labels M_x

Correctness of Step B of the Reachability Function: By construction, $G_{mod_c^x}$ is composed of projections of subsystems G_z, where $z \in S$ and $S = B_c^x$. As a reminder, the states of the indeterminate cycle associated with SEQ_x in G_{d_i} are marked with the label M_x and by construction the states of the machines G_{d_S} and $G_{mod_c^x}$ are marked with labels M_x if one or more state components are marked with the label M_x. Consider any two arbitrarily long traces $\omega_x, \omega_x' \in \mathcal{L}(G_i)$ such that: (i) $f_m \in \omega_x$ where f_m corresponds to the fault type associated with SEQ_x; (ii) $f_m \notin \omega_x'$; (iii) ω_x, ω_x' lead to the indeterminate cycle associated with SEQ_x in G_{d_i} (where states Q^x are labeled M_x); and (iv) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x')$. If there does not exist in $G_{mod_c^x}$ a cycle of states labeled M_x then, by construction of $G_{mod_c^x}$, $\omega_S, \omega_S' \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S') = \omega_x'$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long. Hence, by Lemma 3 and Remark 3, the indeterminate cycle associated with SEQ_x is "Not Reachable" and we return to MDA. If, in Step B, there exists in $G_{mod_c^x}$ a cycle of states labelled M_x then we cannot conclude on the reachability of the indeterminate cycle; thus we number $s_1^c, s_2^c, \ldots, s_P^c$ the sequences of events that describe such cycles and go to Step C.
Correctness of Step C of the Reachability Function: Consider any two arbitrarily long traces $\omega_x, \omega_x' \in \mathcal{L}(G_i)$ such that: (i) $f_m \in \omega_x$ where f_m corresponds to the fault type associated with SEQ_x; (ii) $f_m \notin \omega_x'$; (iii) ω_x, ω_x' lead to the indeterminate cycle associated with SEQ_x in G_{d_i}; and (iv) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x') = sSEQ_x s1$ where $SEQ_x = s_1 s_2$. From equations (12) and (16), we have that $\tilde{s}^x = P_{\{\Sigma_{o_i},\Sigma_{CM_{o_i}}\}}(SEQ_x)$ and, $\forall p \in \{1, \ldots, P\}$, $\tilde{s}_p^c = P_{\{\Sigma_{CM_{oS}},\Sigma_{CM_{o_i}}\}}(s_p^c)$. If $p \in \{1, \ldots, P\}$ such that $\tilde{s}^x = \tilde{s}_p^c$ and if s'^x, s''^x, and $p \in \{1, \ldots, P\}$ such that $\tilde{s}^x = s'^x s''^x$ and $s'^x s''^x = \tilde{s}_p^c$, then, by construction of $G_{mod_c^x}, \omega_S, \omega_S' \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S') = \omega_x'$, and $P_{\{s,o_i\}}(\omega_S)$ is arbitrarily long. Hence, by Lemma 3 and Remark 3, the indeterminate cycle associated with SEQ_x is "Not Reachable" and we return to MDA. If $\exists p \in \{1, \ldots, P\}$ such that $\tilde{s}^x = \tilde{s}_p^c$ or if $\exists s'^x, s''^x$, and $p \in \{1, \ldots, P\}$ such that $\tilde{s}^x = s'^x s''^x$ and $s'^x s'^x = \tilde{s}_p^c$, then we cannot conclude on the reachability of the indeterminate cycle and go to Step D.
Correctness of Step E of the Reachability Function: Consider any two arbitrarily long traces $\omega_x, \omega_x' \in \mathcal{L}(G_i)$ as defined above in the proof of correctness of Step C. Consider $G_{mod_{c-1}^x}$, which is composed of projections of subsystems G_z, where $z \in S$ and $S = B_{c-1}^x$. Since there exist in $G_{mod_{c-1}^x}$ one or

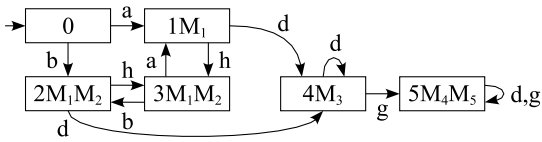


Fig. 6 G_{CM_1}

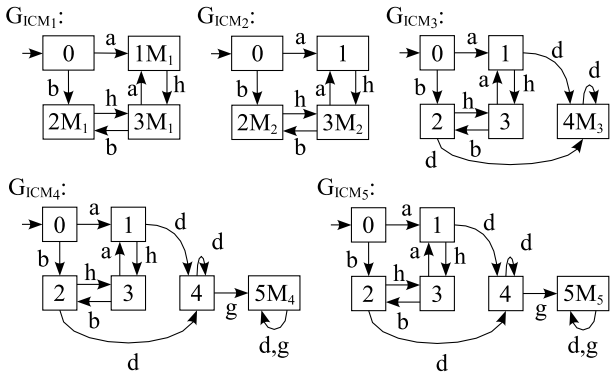


Fig. 7 $G_{CM_2} - G_{CM_5}$ built from G_{CM_1}

more cycles of states labeled M_x that correspond to a projection of SEQ_x, then, by construction of $G_{mod^x_{c-1}}$, $\exists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ that violate the modular diagnosability of $\mathcal{L}(G_S)$ w.r.t. $(\Sigma_{o_z} : z \in S)$ and f_m, and moreover satisfy the following conditions: (i) $f_m \in \omega_S$ where f_m is the fault associated with SEQ_x; (ii) $f_m \notin \omega'_S$; (iii) $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$; (iv) $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$; (v) $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S) = P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega'_S) = sSEQ_x s_1$, $SEQ_x = s_1 s_2$, $x \in \{1, \ldots, X_i\}$; and (vi) $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long. Therefore hypothesis (i) of Lemma 4 is satisfied. The condition $S_c = \emptyset$ implies that there does not exist any subsystem G_l, $l \notin S$, that contains common events with the automaton $\widetilde{G}_{c-1}$; thus hypothesis (ii) of Lemma 4 is satisfied. Then, by Lemma 4 and Remark 4, we declare the indeterminate cycle associated with SEQ_x "Reachable" in G_S and G_T and return to MDA. If $S_c \neq \emptyset$ then we cannot decide on the reachability of the indeterminate cycle (cf. Lemma 5) and go to Step B.

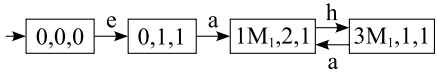
We have proven that the Reachability Function returns the correct answer to the question: "Is the indeterminate cycle associated with SEQ_x reachable in the global system behavior $\mathcal{L}(G_T)$?." We use this to complete the proof of the correctness of MDA.

Correctness of Step 1 of MDA: The correctness of Step 1 follows directly from Corollary 1.

Correctness of Step 2-d of MDA: If, in the Reachability Function, we declare the indeterminate cycle associated with SEQ_x "Reachable" then we conclude that, by Lemma 4, $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}, which also implies that $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.

Correctness of Step 3 of MDA: If, in the Reachability Function, we declare the indeterminate cycles associated with SEQ_x, $x = 1, \ldots, X_i$, "Not Reachable" then, by Corollary 2, we conclude that $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}. If the above is true for all $i \in ND$ then, by Lemma 1, $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. ■

Fig. 8 $G_{mod^1_2}$



Discussion

To give more insight into MDA, we present its flowchart, cf. Fig. 9, and discuss the key steps of its operation. The procedure starts by building local diagnosers for each module of the system and checking if they are monolithically diagnosable or not. As mentioned before, if each individual module is (monolithically/modularly) diagnosable, then the complete system is both monolithically and modularly diagnosable. Therefore, we need only focus on the modules that are not diagnosable in order to find out if a violation of modular diagnosability occurs or not when the given module is coupled with the rest of the system.

To do so, we concentrate on the traces that form indeterminate cycles in local diagnosers, called the troublesome traces (cf. the Preliminary Discussion section). We need to test these troublesome traces one by one and determine if they survive in the diagnoser of the complete system, without necessarily constructing this monolithic diagnoser.

The testing procedure starts by selecting one indeterminate cycle in a given (non-diagnosable) local diagnoser and isolating all its troublesome traces (there could be more than one troublesome trace depending on the accessibility of the indeterminate cycle in the transition structure of the local diagnoser). For each troublesome trace, we select all other modules that contain an event common with the ones in the troublesome trace, build observer automata for common events (cf. Step 2-b of Algorithm 1) for each module selected, perform the parallel composition of these automata, and finally check if the indeterminate cycle under consideration survives (cf. Step B of Algorithm 3). If it does not survive at this stage then it will not survive if we were to construct the monolithic diagnoser. However, if it does survive, then we need to consider the effect of other modules, namely those that have common events with the result of the above parallel composition. This is the heart of the incremental procedure performed in Algorithm 3. We iterate using essentially the same steps as described above—cf. the loop formed by Steps B through E of Algorithm 3.

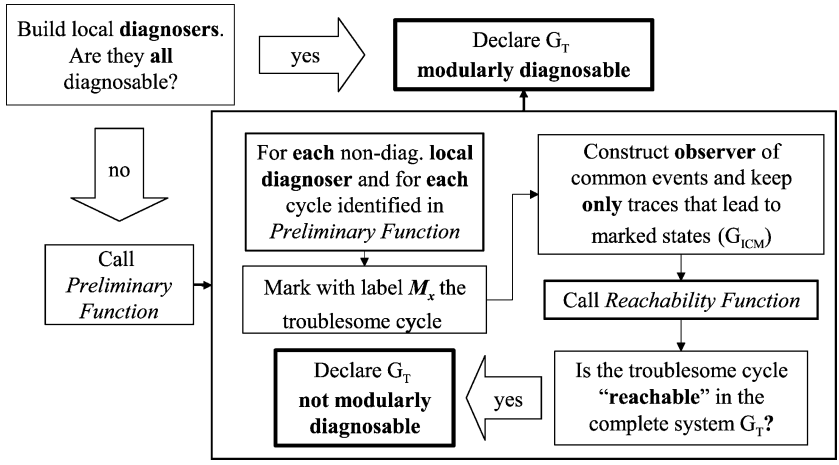


Fig. 9 Algorithm flowchart

The incremental procedure in Algorithm 3 ceases to add local modules and consequently stops when either (i) it has been determined that the indeterminate cycle under consideration is not reachable in the complete system—if this holds for all indeterminate cycles then the monolithic system is modularly diagnosable or (ii) no other module is added in the incremental process at Step E of Algorithm 3—in which case the monolithic system is not modularly diagnosable. Note that the latter conclusion can be reached *without* having to consider all modules in the set T. This potential computational gain depends on the structure of the automaton $G_{mod_c^x}$ and its co-accessibility properties with respect to the indeterminate cycle under consideration, as determined in Steps C and D of Algorithm 3.

Figure 10 describes the architecture of the modular diagnosability decision process with respect to Module 1. The process has to be repeated for all modules in the system in order to infer on the modular diagnosability of the monolithic system.

The main feature exploited within MDA is the incremental addition of modules by considering only those that are necessary to reach a decision on the modular diagnosability of the monolithic system. Depending on the structure of the system, MDA may consider a smaller number of modules rather than all of T when performing parallel composition operations. The worst case can possibly occur and yield $|B_c^x| = |T|$, which implies that every system module is considered in the parallel composition for obtaining $G_{mod_c^x}$. At this stage, further computational experience is needed to more precisely assess the role of the system's structure on the computational complexity of MDA.

The whole procedure followed in MDA not only exploits the modular structure of the given system, but also may provide insight into causes of non diagnosability and possible remedies for it through coupling of system modules with one another. Thus MDA could be a useful tool in modular system design.

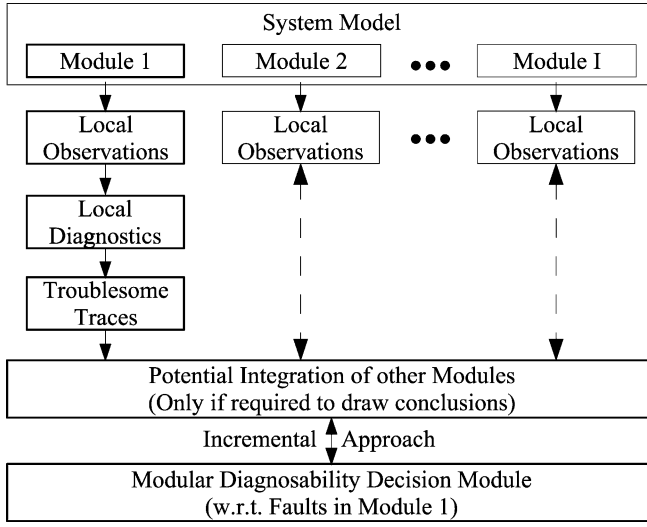


Fig. 10 Modular diagnosability verification

Online Diagnosis

If the system G_T, $T = \{1, \ldots, I\}$, is modularly diagnosable, we can perform *online diagnosis* by simply running the local diagnosers G_{d_i}, $i \in T$, at each local site, cf. Fig. 11. We know from the property of modular diagnosability that even if the local diagnoser G_{d_i} contains an indeterminate cycle, the local observations at site i will never stay forever in this cycle when the complete system G_T is functioning.

If MDA outputs that the system is not modularly diagnosable, then we can still partially diagnose the system online as follows. Each indeterminate cycle is associated to a fault $f_m \in \Sigma_{f_T}$, $m \in \{1, \ldots, M\}$. From MDA, we know which indeterminate cycles of G_{d_i} are reachable and which are blocked. If the local diagnoser G_{d_i} contains an indeterminate cycle that is "reachable" in the complete system G_T, then local observations may stay forever in this cycle. Therefore we mark as "f_m inactive" the states of G_{d_i} that correspond to the reachable indeterminate cycle associated to the fault f_m. We run at each local site the modified version of the local diagnoser G_{d_i}, i.e., the one with the labels "f_m inactive." Then, when a local diagnoser reaches an "f_m inactive" state, the local site broadcasts that there is a potential fault f_m that cannot be diagnosed with certainty.

Conclusion

We have proposed a notion of modular diagnosability that is suitable for systems that have modular structure expressed in terms of the parallel composition of individual automata, where each individual automaton models the behavior of the system component at the corresponding site. If modular diagnosability holds, then on-line fault diagnosis of the modular system is straightforward as it suffices to run a local diagnoser at each site, where the local diagnoser is built using only the local automaton model and ignoring the remainder of the system model. It is guaranteed that, after sufficiently many local observable events, any fault at a site will be diagnosed. However, the verification of modular diagnosability requires in general the joint consideration of multiple system

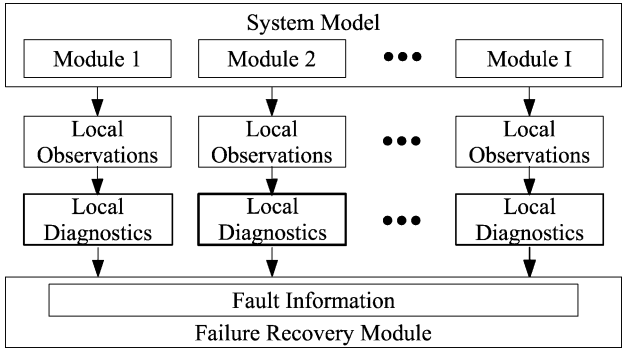


Fig. 11 Online modular diagnosis

components. We have presented an algorithm that correctly verifies if modular diagnosability holds or not and does so by incrementally including the automata models of other system components only if they are required to draw definitive conclusions about the diagnosability of faults within a given system component. This property of the algorithm makes it potentially computationally advantageous for large complex modular systems. Moreover, even if the modular diagnosability property does not hold, the algorithm provides insight into possible structural changes to the system in order to render it modularly diagnosable.

Acknowledgements This research is supported in part by NSF grants ECS-0080406, CCR-0082784, and CCR-0325571, by ONR grant N00014-03-1-0232, and by a grant from the Xerox University Affairs Committee. The authors are grateful to the reviewers for their pertinent comments and for pointing out an error in an example. They also acknowledge useful discussions with Rami Debouk.

Appendix

For system $G = (X, \Sigma, \delta, x_0)$, $\Sigma_o \subseteq \Sigma$, and fault types $\{F_1, F_2, \ldots, F_m\}$, we recall the following definitions originally introduced in Sampath et al. (1995).

- Diagnoser (G_d)

$$G_d = (Q_d, \Sigma_o, \delta_d, q_0) \tag{20}$$

$$\Delta = \{N\} \cup 2^{\Delta_F}, \text{ where } \Delta_F = \{F_1, F_2, \ldots, F_m\} \tag{21}$$

$$X_o = \{x_o\} \cup \{x \in X : x \text{ has an observable event into it}\} \tag{22}$$

$$\delta_{\mathrm{d}} : \text{ transition function of the diagnoser} \tag{23}$$

$$q_{\mathrm{o}} = \{(x_o, \{N\})\} \tag{24}$$

$$Q_{\mathrm{o}} = 2^{X_o \times \Delta} \tag{25}$$

$$Q_d : \text{ subset of } Q_o \text{ reachable under } \delta_d \tag{26}$$

- Non-deterministic automaton without unobservable events (G')

$$G' = (X_o, \Sigma_o, \delta_{G'}, x_0), \tag{27}$$

where

$$\mathcal{L}(G') = P(L) = \{t : t = P(s) \text{ for some } s \in L\}. \tag{28}$$

The elements X_o, Σ_o, and x_0 are as defined above. The transition relation of G' is given by

$$\delta_{G'} \subseteq (X_o \times \Sigma \times X_o) \tag{29}$$

and is defined as follows

$$(x,\ \sigma,\ x') \in \delta_{G'} \text{ if } \delta(x,\ s) = x' \text{ for some } \mathrm{s} \in L_\sigma(G,\ x). \tag{30}$$

References

Aghasaryan A, Fabre E, Benveniste A, Boubour R, Jard C (1998, June). Fault detection and diagnosis in distributed system: an approach by partially stochastic Petri nets. Discret Event Dyn. Syst. Theory Appl. 8(2):203–231.

Benveniste A, Fabre E, Haar S, Jard C, (2003, May). Diagnosis of asynchronous discrete event systems, a net unfolding approach. IEEE Trans Autom Contr 48(5):714–727.

Bouloutas AT, Calo S, Finkel A, (1994, Feb/Mar/Apr). Alarm correlation and fault identification in communication networks. IEEE Trans Commun 42(2/3/4):523–533.

Cassandras CG. Lafortune S (1999). Introduction to Discrete Event Systems. Kluwer Academic Publishers.

Console L, Picarsi C, Ribaudo M (2002). Process algebras for system diagnosis. Artif Intell 142(1):19–51.

Contant O, Lafortune S, Teneketzis D, (2004a, November). Diagnosis of distributed discrete event systems: architecture and modular verification approach. Technical Report CGR 04-04, College of Engineering Control Group Reports, University of Michigan, April. Revised.

Contant O, Lafortune S, Teneketzis D, (2004b, April). Diagnosis of intermittent faults. Discret Event Dyn Syst Theory Appl 14(2):171–202.

Coolen R, Luiijf H (2002, January). Intrusion detection: generics and state-of-the-art. Technical Report RTO-TR-049, Research and Technology Organisation, NATO, Neuilly-sur-Seine, France.

Debouk R, Lafortune S Teneketzis D (2000, January). Coordinated decentralized protocols for failure diagnosis of discrete events systems. Discret Event Dyn Syst Theory Appl 10(1–2):33–86.

Debouk R, Malik R, Brandin B (2002, December). A modular architecture for diagnosis of discrete event systems. Proc. 41st IEEE Conf. on Decision and Control—CDC'02, Las Vegas, NV, USA, pp 417–422.

Garcia E, Morant F, Blasco-Giménez R, Correcher A, Quiles E, (2002, October). Centralized modular diagnosis and the phenomenon of coupling. Proc. 2002 IFAC International Workshop on Discrete Event Systems—WODES'02, Zaragoza, Spain, pp 161–168.

Genc S, Lafortune S (2003, June). Distributed diagnosis of discrete-event systems using Petri nets. Proc. 2003 International Conf. on Applications and Theory of Petri Nets, Eindhoven, The Netherlands, pp 316–336.

Hashtrudi Zad S, Kwong RH, Wonham WM, (2003, July). Fault diagnosis in discrete-event systems: framework and model reduction. IEEE Trans Autom Contr 48(7):1199–1212.

Holloway LE, Chand S (1994, June). Time templates for discrete event fault monitoring in manufacturing systems. Proc. 1994 American Control Conference—ACC'94, Baltimore, MD, USA, pp 701–706.

Jiang S, Kumar R (2002, May). Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. Proc. 2002 American Control Conference— ACC'02, Anchorage, AK, USA, pp 128–133.

Jiang S, Kumar R, Garcia HE (2003, April). Diagnosis of repeated/intermittent failures in discrete event systems. IEEE Trans Robot Autom 19(2):310–323.

Lafortune S, Teneketzis D, Sampath M, Sengupta R, Sinnamohideen K (2001, June). Failure diagnosis of dynamis systems: an approach based on discrete event systems. Proc. 2001 American Control Conference—ACC'01, Arlington, VA, USA, pp 2058–2071.

Lamperti G, Zanella M (2003). Diagnosis of active systems: principles and techniques, volume 741 of the Kluwer International Series in Engineering and Computer Science. Dordrecht, NL Kluwer Academic Publishers.

Lamperti G, Zanella M (2004, October). A bridged diagnosis method for the monitoring of polymorphic discrete-event systems. IEEE Trans Syst Man Cybern, Part B 34(5):2222–2244.

Lin F (1994, May). Diagnosability of discrete event systems and its applications. Discret Event Dyn Syst Theory Appl. 4(2):197–212.

Lunze J (2000). Diagnosis of quantized systems based on a timed discrete-event model. IEEE Trans Syst Man Cybern, Part A 30(3):322–335.

Pandalai DN, Holloway LE (2000, May). Template languages for fault monitoring of discrete event processes. IEEE Trans Autom Contr 45(5):868–882.

Pencolé Y (2000). Decentralized diagnozer approach: application to telecommunication networks. In: Provan G Darwiche A (eds) Proc. of the 11th International Workshop on Principles of Diagnosis—DX'00, Morelia, Mexico, pp 185–192.

Pencolé Y, Cordier M-O Rozé L (2002). A decentralized model-based diagnostic tool for complex systems. Int J Artif Intell Tools 11(3):327–346.
Ricker LS, Fabre E (2000, December). On the construction of modular observers and diagnosers for discrete event systems. Proc. 39th IEEE Conf. on Decision and Control—CDC'00, Sydney, Australia, pp 2240–2244.
Sampath M (2001, June). A hybrid approach to failure diagnosis of industrial systems. In Proc. 2001 American Control Conference—ACC'01, Arlington, VA, USA.
Sampath M, Sengupta R, Sinnamohideen K, Lafortune S, Teneketzis D (1995, September). Diagnosability of discrete event models. IEEE Trans. Contr. Syst. Technol. 40(9):1555–1575.
Sampath M, Sengupta R, Sinnamohideen K, Lafortune S, Teneketzis D (1996, March). Failure diagnosis using discrete event systems. IEEE Trans Contr Syst Technol 4(2):105–124.
Sampath M, Lafortune S, Teneketzis D (1998, July). Active diagnosis of discrete event systems. IEEE Trans Autom Contr 43(7):908–929.
Sengupta R (2001, June). Discrete-event diagnostics of automated vehicles and highways. Proc. 2001 American Control Conference—ACC'01. Arlington, VA, USA, pp 25–27.
Sinnamohideen K (2001, June). Discrete-event diagnostics of heating, ventilation, and air-conditoning systems. Proc. 2001 American Control Conference—ACC'01, Arlington, VA, USA.
Su R, Wonham WM (2004, September). Hierarchical distributed diagnosis under global consistency. In Proc. 2004 IFAC International Workshop on Discrete Event Systems—WODES'04. Reims, France, pp 157–162.
Su R, Wonham WM, Kurien J, Koutsoukos X (2002, October). Distributed diagnosis for qualitative systems. Proc. 2002 IFAC International Workshop on Discrete Event Systems—WODES'02, Zaragoza, Spain, pp 169–174.
Yoo T-S Garcia HE (2004, June). Event diagnosis of discrete-event systems with uniformly and nonuniformly bounded diagnosis delays. Proc. 2004 American Control Conference—ACC'04. Boston, MA, USA, pp 5102–5107.