Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 521–530

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Temporal-Fault Diagnosis for Critical-Decision Making in Discrete-Event Systems

Nicola Bertoglio[a], Gianfranco Lamperti[a,*], Marina Zanella[a], Xiangfu Zhao[b]

[a]*Department of Information Engineering, University of Brescia, 25123 Brescia, Italy*
[b]*School of Computer and Control Engineering, Yantai University, 264005 Yantai, China*

Abstract

Since its appearance in AI, model-based diagnosis is intrinsically set-oriented. Given a sequence of observations, the diagnosis task generates a set of diagnoses, or *candidates*, each candidate complying with the observations. What all the approaches in the literature have in common is that a candidate is invariably a *set* of faulty elements (components, events, or otherwise). In this paper, we consider a posteriori diagnosis of discrete-event systems (DESs), which are described by networks of components that are modeled as communicating automata. The diagnosis problem consists in generating the candidates involved in the trajectories of the DES that conform with a given temporal observation. Oddly, in the literature on diagnosis of DESs, a candidate is still a *set* of faulty events, despite the temporal dimension of trajectories. In our view, when dealing with critical domains, such as power networks or nuclear plants, set-oriented diagnosis may be less than optimal in explaining the supposedly abnormal behavior of the DES, owing to the lack of any temporal information relevant to faults, along with the inability to discriminate between single and multiple occurrences of the same fault. Embedding temporal information in candidates may be essential for critical-decision making. This is why a temporal-oriented approach is proposed for diagnosis of DESs, where candidates are *sequences* of faults. This novel perspective comes with the burden of unbounded candidates and infinite collections of candidates, though. To cope with, a notation based on regular expressions on faults is adopted. The diagnosis task is supported by a *temporal diagnoser*, a flexible data structure that can grow over time based on new observations and domain-dependent *scenarios*.

Keywords: model-based diagnosis; discrete-event systems; temporal observations; temporal faults; communicating automata; temporal diagnoser; behavioral scenarios; incremental techniques

1. Introduction

As the name suggests, model-based diagnosis [19, 10] aims to find out the causes of the supposedly faulty behavior of a system based on its model. When the system is static, like a combinational circuit, the (behavioral) model of a

* Gianfranco Lamperti. Tel.: +39-030-371-5491 ; fax: +39-030-380-014.
E-mail address: gianfranco.lamperti@unibs.it

1877-0509
10.1016/j.procs.2020.08.054

component (for instance, a logical gate) is represented by a mapping from the possible input to the expected output. In a dynamical system, time comes into play and model-based diagnosis needs to account for the state of the system too [22]. For various reasons, a dynamical system may be conveniently modeled as a discrete-event system (DES) [8], where the DES changes state over discrete time. However, the way the DES is represented is varying, typically as a monolithic finite-state machine [17], as a network of components [15], each of them being modeled as a communicating automaton [6], or as a Petri net [2, 9]. Although the automaton of a DES component can represent just the nominal (correct) behavior [18], usually each state transition is either *normal* or *abnormal*, as in the seminal work by [20, 21].

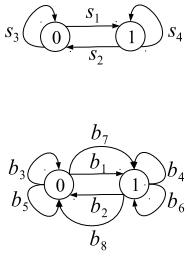
In the literature, diagnosing a DES amounts to generating a set of *candidates* based on a given *temporal observation*, namely a sequence of temporally-ordered observations, where a candidate is a set of *faults*, with each fault being an abnormal event or transition. The diagnosis of a DES is a form of *abductive* reasoning, as the candidates are generated based on the trajectories (sequences of state transitions) of the DES that conform with the temporal observation. The classical approach in [20] performs the abduction offline, by compiling the DES models into a *diagnoser*, a data structure that is exploited online in order to produce a new set of candidates upon the reception of each new observation (*monitoring-based diagnosis*). By contrast, in the *active-system* approach [1, 15], the abduction can be carried out online by focusing on the trajectories that conform with the temporal observation. Specifically, the diagnosis output is the set of candidates relevant to the (possibly infinite) set of trajectories of the DES that produce the temporal observation. Since the domain of faults is finite, both the candidates and the diagnosis output are finite and bounded.

Still, in both the diagnoser approach and the active-system approach, a candidate is a *set* of faults. This sounds strange considering that a candidate is the projection of a trajectory, which is not a set, but a *sequence* of state transitions. The point is, since a set is a collection of unordered elements without duplicates, both the temporal ordering of faults and their multiple occurrences are completely lost within a set. When the diagnosis task focuses on critical systems or processes, such as a large power network, a nuclear plant, or even the evolution of a pandemic like the current COVID-19, set-orientated diagnosis may be less than optimal in explaining a supposedly abnormal behavior, owing to the lack of any temporal information relevant to faults, along with the inability to discriminate between single and multiple occurrences of the same fault. Hence, embedding temporal information in candidates may be essential for critical-decision making.

To this end, we introduce the notion of a *temporal fault*, which is a sequence of temporally-ordered faults associated with a trajectory of the DES. Despite the preservation of the temporal information in a candidate, the act of replacing a sequence for a set comes with two problems, however: (1) the length of candidates is possibly unbounded, and (2) the set of candidates may be infinite. Fortunately, the set of candidates that explain a temporal observation turns out to be a regular language on faults, thus it can be represented as a regular expression. In other words, both the unboundedness of candidates and the infinity of the set of candidates can be concisely represented by regular expressions on faults.

Given the notion of a temporal fault, a diagnosis technique is proposed based on a data structure (namely, a finite automaton) called a *temporal diagnoser*, which allows for the efficient a posteriori diagnosis of a DES. A temporal diagnoser is generated offline in its entirety, so that any temporal observation can be explained online efficiently. This approach is only ideal, however, as in real DESs the size of a temporal diagnoser grows exponentially with the number of components in the DES. Although a temporal diagnoser retains legitimacy as a formal reference (as is for the diagnoser in [20]), a more practical approach is presented in the next sections, which introduce a *partial temporal diagnoser*, this being initially generated offline as a prefix of the temporal diagnoser.

Remarkably, given a temporal observation that is not included in the language of the partial temporal diagnoser, the latter can be extended online while solving the relevant diagnosis problem. Moreover, a partial temporal diagnoser can be enriched based on domain-dependent behavioral *scenarios* [3, 4, 5]. A scenario is a concise way to specify a collection of evolutions of the DES (e.g. all the trajectories that include three occurrences of a specific transition or all the trajectories that are affected by a single fault). This notion comes in handy to specify the (interesting and/or critical and/or frequent) evolutions to be added to the partial temporal diagnoser. In recent work by the authors, scenarios were applied to both monitoring [4, 5] and a posteriori diagnosis of DESs [3], the latter being the same task that is dealt with in this paper. However, while a set-oriented perspective on diagnosis results is adopted in [3], a new temporal-oriented approach is proposed in this paper.



t	*Description*	o	f
s_1	The sensor detects a threatening external event *ko* and generates the open event	*sen*	ε
s_2	The sensor detects a liberating external event *ok* and generates the close event	*sen*	ε
s_3	The sensor detects a threatening external event *ko*, yet generates the close event	ε	$\mathbf{f_1}$
s_4	The sensor detects a liberating external event *ok*, yet generates the open event	ε	$\mathbf{f_2}$
b_1	The breaker reacts to the open internal event by opening	*bre*	ε
b_2	The breaker reacts to the close internal event by closing	*bre*	ε
b_3	The breaker does not react to the open internal event and remains closed	ε	$\mathbf{f_3}$
b_4	The breaker does not react to the close internal event and remains open	ε	$\mathbf{f_4}$
b_5	The breaker reacts to the close internal event by remaining closed	*bre*	ε
b_6	The breaker reacts to the open internal event by remaining open	*bre*	ε
b_7	The breaker reacts to the close internal event by opening	*bre*	$\mathbf{f_5}$
b_8	The breaker reacts to the open internal event by closing	*bre*	$\mathbf{f_6}$

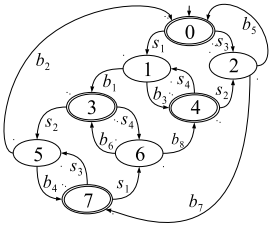


Fig. 1. Models of sensor and breaker of DES $\mathcal{P}$ (left), details of component transitions (table in the center), and space $\mathcal{P}^*$ (right).

2. Modeling a DES

A DES $\mathcal{X}$ is a network of *components*, where the behavior of each component is modeled as a communicating automaton [6]. A component is endowed with input and output *pins*, where each output pin is connected with an input pin of another component by a *link*. A component reacts to events (occurring either outside the DES or inside it) by performing a transition. When performing a transition, a component consumes the triggering (input) event and possibly generates new events on its output pins, which are bound to trigger the transitions of other components. This results in a sequence of component transitions, called a *trajectory* of $\mathcal{X}$, at the end of which $\mathcal{X}$ becomes *idle*, namely without any internal event to be consumed. A contiguous subsequence of a trajectory is called a *trajectory segment*. At the occurrence of a component transition, $\mathcal{X}$ changes its state, with a state of $\mathcal{X}$ being a pair of the array of the current component states and the array of the (possibly empty) current events placed in links. Formally, the (possibly infinite) set of trajectories of $\mathcal{X}$ is specified by a deterministic finite automaton (DFA), namely the *space* $\mathcal{X}^*$ of $\mathcal{X}$, $\mathcal{X}^* = (\Sigma, X, \tau, x_0, X_\mathrm{f})$, where Σ (the alphabet) is the set of component transitions, X is the set of states, τ is the deterministic transition function mapping a state and a component transition into a new state, $\tau : X \times \Sigma \mapsto X$, x_0 is the initial state, and X_f is the set of final (idle) states. For diagnosis purposes, the model of $\mathcal{X}$ needs to be enriched with a *mapping table*. Let $\mathbf{T}$ be the set of component transitions in $\mathcal{X}$, $\mathbf{O}$ a finite set of *observations*, and $\mathbf{F}$ a finite set of *faults*. The *mapping table* μ of $\mathcal{X}$ is a function $\mu(\mathcal{X}) : \mathbf{T} \mapsto (\mathbf{O} \cup \{\varepsilon\}) \times (\mathbf{F} \cup \{\varepsilon\})$, where ε is the *empty* symbol. The table $\mu(\mathcal{X})$ can be represented as a finite set of triples (t, o, f), where $t \in \mathbf{T}$, $o \in \mathbf{O} \cup \{\varepsilon\}$, and $f \in \mathbf{F} \cup \{\varepsilon\}$. The triple (t, o, f) defines the observability and normality of t: if $o \neq \varepsilon$, then t is *observable*, else t is *unobservable*; likewise, if $f \neq \varepsilon$, then t is *faulty*, else t is *normal*. Based on $\mu(\mathcal{X})$, each trajectory T in $\mathcal{X}^*$ can be associated with a *temporal observation*. The *temporal observation* of T is the sequence of observations involved in T, $Obs(T) = [\, o \mid t \in T, (t, o, f) \in \mu(\mathcal{X}), o \neq \varepsilon \,]$. In the literature, a trajectory T is also associated with a diagnosis, namely the set of faults involved in T. As such, a diagnosis does not indicate the temporal relationships among faults, nor does it account for multiple occurrences of the same fault. On the other hand, treating a diagnosis as a set of faults guarantees that the domain of possible diagnoses is finite, being bounded by the powerset of the domain of faults. In contrast with this classical perspective, we introduce the notion of a temporal fault, which, in our view, better supports critical decision-making.

Definition 1. *Let T be a trajectory of a DES. The* temporal fault *of T is the sequence of faults in T, $Flt(T) = [\, f \mid t \in T, (t, o, f) \in \mu(\mathcal{X}), f \neq \varepsilon \,]$.*

Since the length of T is in general unbounded, so too is in general the length of both $Obs(T)$ and $Flt(T)$. A contiguous subsequence of a temporal fault is called a *temporal-fault segment*.

Example 1. With reference to Figure 1, we consider a DES $\mathcal{P}$ (protection) that includes two components, a sensor s and a breaker b, and one link from s to b. The models of s and b are outlined on the left of the figure, where circles and arcs denote states and transitions, respectively. Each component transition is described in the table. The observations o and faults f associated with the component transitions, namely the mapping table $\mu(\mathcal{P})$, are listed on the right side of the table. Only one observation is provided for both the sensor and the breaker, namely *sen* and *bre*, respectively, each being associated with several transitions. Six faults are defined as follows. $\mathbf{f_1}$: the sensor commands the breaker to close rather than to open; $\mathbf{f_2}$: the sensor commands the breaker to open rather than to close; $\mathbf{f_3}$: the breaker remains closed

instead of opening; $\mathbf{f_4}$: the breaker remains open instead of closing; $\mathbf{f_5}$: the breaker opens instead of remaining closed; $\mathbf{f_6}$: the breaker closes instead of remaining open. The space of $\mathcal{P}$, namely $\mathcal{P}^*$, is depicted on the right side of the figure, where each state is identified by a number (details of component states and events are omitted), with 0 being the initial state and $\{0, 3, 4, 7\}$ being the set of final states (double circled). Owing to cycles, the set of possible trajectories of $\mathcal{P}$ is infinite, one of them being $T = [s_3, b_5, s_1, b_3, s_4, b_3, s_2, b_5]$. Based on $\mu(\mathcal{P})$, we have $Obs(T) = [bre, sen, sen, bre]$ and $Flt(T) = [\mathbf{f_1}, \mathbf{f_3}, \mathbf{f_2}, \mathbf{f_3}]$. In a set-oriented perspective, the diagnosis of T would be the set $\{\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}\}$, rather than a sequence, where neither the temporal ordering of faults nor the double occurrence of $\mathbf{f_3}$ is embedded.

3. Temporal Diagnosis

The goal in diagnosing a DES is generating the set of candidates relevant to a temporal observation O. In contrast with the classical set-oriented perspective, here a candidate is a temporal fault that is produced by a trajectory that entails O. The (possibly infinite) set of candidates is the *temporal diagnosis* of O, as formalized in Definition 2.

Definition 2. *Let O be a temporal observation of $\mathcal{X}$. The* temporal diagnosis *of O is the set of temporal faults relevant to the trajectories that conform with O, namely $\Delta(O) = \{\, Flt(T) \mid T \in \mathcal{X}^*, Obs(T) = O \,\}$.*

A temporal diagnosis may include an infinite number of temporal faults. Still, any temporal diagnosis can be concisely represented by a regular expression on a set of faults, as shown in the next example.

Example 2. Let $O = [bre, sen]$ be a temporal observation of the DES $\mathcal{P}$ defined in Example 1. Based on the space $\mathcal{P}^*$ and the observations associated with the component transitions and defined in Figure 1, the language of the trajectories generating O can be represented by a regular expression, namely $s_3 b_5 s_1 b_3 (s_4 b_3)^*$. Hence, $\Delta(O) = \mathbf{f_1 f_3 (f_2 f_3)}^*$, which is represented by a regular expression on the faults $\mathbf{f_1}$, $\mathbf{f_2}$, and $\mathbf{f_3}$. In a classical set-oriented perspective, the set of candidates would be $\{\{\mathbf{f_1}, \mathbf{f_3}\}, \{\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}\}\}$, in which case we know that both fault $\mathbf{f_1}$ and $\mathbf{f_3}$ have certainly occurred, whereas the occurrence of fault $\mathbf{f_2}$ is uncertain. Besides, we have no hint about how many times such faults have manifested themselves and in which temporal order. If, instead, the regular expression is given, we know for sure that fault $\mathbf{f_1}$ has occurred just once and it was the first, while $\mathbf{f_3}$ was the second. In addition, we know that, if $\mathbf{f_2}$ has occurred, then it has occurred after them, and it may have occurred several times, every time being followed by $\mathbf{f_3}$. All these details may be essential to understand what has happened inside the system in order to make critical decisions, including appropriate recovery actions.

In order to support the efficient generation of the temporal diagnosis of any temporal observation, a *temporal diagnoser* is introduced. The temporal diagnoser acts somewhat as a counterpoint to the *diagnoser* data structure which is exploited in classical set-oriented diagnosis of DESs [20, 21].

4. Temporal Diagnoser

The temporal diagnoser of a DES $\mathcal{X}$ is an NFA resulting from the (offline) compilation of $\mathcal{X}$. The alphabet of the temporal diagnoser is a set of triples $(o, \mathcal{L}, f)$, where o is an observation of $\mathcal{X}$, $\mathcal{L}$ is a language on the faults of $\mathcal{X}$, and f is a (possibly empty) fault. Roughly, each state of the temporal diagnoser (namely a *fault space*) embodies a sort of local explanation defined by languages (in fact, regular expressions) on faults.

Definition 3. *Let $\mathcal{X}^*$ be the space of $\mathcal{X}$ having mapping table $\mu(\mathcal{X})$, $\mathbf{F}$ the set of faults of $\mathcal{X}$, and $\bar{x}$ a state in $\mathcal{X}^*$. The* fault space *of $\bar{x}$ is an NFA $\mathcal{X}^*_{\bar{x}} = (\Sigma, X, \tau, x_0, X_\mathrm{f})$, where $\Sigma = \mathbf{F} \cup \{\varepsilon\}$ is the alphabet, X is the subset of the states of $\mathcal{X}^*$ that are reachable from $\bar{x}$ by unobservable transitions, $x_0 = \bar{x}$ is the initial state, X_f is the set of final states (the states that are final in $\mathcal{X}^*$), and $\tau : X \times \Sigma \mapsto 2^X$ is the transition function, where $\langle x_1, f, x_2 \rangle$ is an arc in τ iff $\langle x_1, t, x_2 \rangle$ is a transition in $\mathcal{X}^*$ and $(t, \varepsilon, f) \in \mu(\mathcal{X})$. Each state $x \in X$ is marked with the language of the temporal-fault segments of the trajectory segments in $\mathcal{X}^*$ from $\bar{x}$ to x, denoted $\mathcal{L}(x)$. The* diagnosis language *of the fault space $\mathcal{X}^*_{\bar{x}}$ is a regular language on the set of faults of $\mathcal{X}$ defined as follows:*

$$\mathcal{L}(X^*_{\bar{x}}) = \begin{cases} \emptyset & \text{if } X_\mathrm{f} = \emptyset \\ \mathcal{L}(x) & \text{if } X_\mathrm{f} = \{x\} \\ \mathcal{L}(x_1) \cup \ldots \cup \mathcal{L}(x_n) & \text{if } X_\mathrm{f} = \{x_1, \ldots, x_n\}. \end{cases} \tag{1}$$

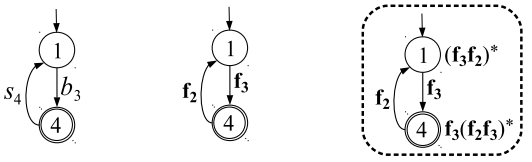


Fig. 2. Genesis of the fault space $\mathcal{P}_1^*$.

Example 3. With reference to the DES $\mathcal{P}$ introduced in Example 1, shown in Figure 2 is the genesis of the fault space $\mathcal{P}_1^*$ based on Definition 3. The graph on the left represents the portion of the space $\mathcal{P}^*$ that is reached by unobservable transitions starting from the state 1. Then, the identifiers of the transitions are replaced with the corresponding faults, thereby obtaining the graph in the center. The actual fault space is depicted on the right of Figure 2, where states 1 and 4 are marked with relevant regular expressions on faults. Based on eqn. (1), we have $\mathcal{L}(\mathcal{P}_1^*) = \mathcal{L}(4) = \mathbf{f_3}(\mathbf{f_2f_3})^*$.

The decoration of the states in the fault space displayed in Figure 2 can be carried out by inspection of the NFA shown in the center of the figure. However, a general technique allowing for the automatic decoration of the states within a fault space is needed. To this end, we have adapted the algorithm proposed in [7] in the context of sequential circuit state diagrams. Essentially, this algorithm takes as input an NFA and generates the regular expression of the language accepted by this NFA. Still, in a fault space, all states need to be marked with the relevant regular expressions. Thus, we have extended the algorithm to decorate all the states in one processing of the NFA (rather than one processing for each state).

Definition 4. *Let $\mathcal{X}^* = (\Sigma, X, \tau, x_0)$ be the space of $\mathcal{X}$, $\mathbf{O}$ the set of observations of $\mathcal{X}$, $\mathbf{F}$ the set of faults of $\mathcal{X}$, and $\mathbf{L}$ the set of regular languages on $\mathbf{F} \cup \{\varepsilon\}$. The* temporal diagnoser *of $\mathcal{X}$ is an NFA $\mathcal{X}^\Delta = (\Sigma', X', \tau', x_0', X_f')$, where $\Sigma' \subseteq \mathbf{O} \times \mathbf{L} \times (\mathbf{F} \cup \{\varepsilon\})$ is the alphabet, X' is the set of states, where each state is a fault space of a state of $\mathcal{X}^*$, $x_0' = \mathcal{X}^*_{x_0}$ is the initial state, X_f' is the set of final states (the fault spaces including at least one final state in $\mathcal{X}^*$), and τ' is the (nondeterministic) transition function, $\tau' : (X' \times X) \times \Sigma' \mapsto 2^{(X' \times X)}$, where $\langle (x_1', x_1), (o, \mathcal{L}(x_1), f), (x_2', x_2) \rangle$ is an arc in τ' iff x_1 is a state in x_1', $\langle x_1, t, x_2 \rangle \in \tau$, $(t, o, f) \in \mu(\mathcal{X})$ with $o \neq \varepsilon$, and $x_2' = \mathcal{X}^*_{x_2}$.*

Example 4. With reference to the DES $\mathcal{P}$, shown in Figure 3 is the temporal diagnoser $\mathcal{P}^\Delta$, where the states (fault spaces) are renamed $0 \cdots 7$. Unlike component transitions within fault spaces, which are represented with plain arcs, the transitions between states of $\mathcal{P}^\Delta$ are depicted as dashed arcs that are marked with the relevant triples.

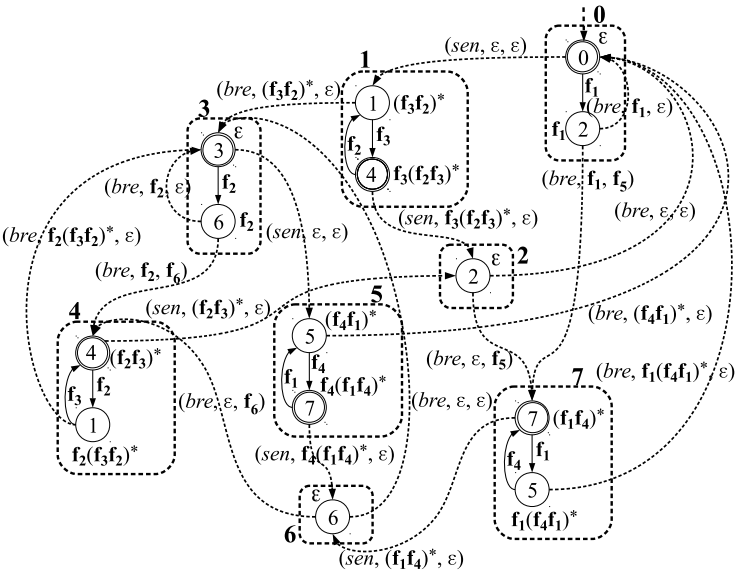


Fig. 3. Temporal diagnoser of the DES $\mathcal{P}$, namely $\mathcal{P}^\Delta$.

Algorithm 1 *Ideal Diagnosis*

1: **procedure** IDEAL DIAGNOSIS($\mathcal{X}^\Delta$, O, $\mathcal{R}$)
2: **input** $\mathcal{X}^\Delta = (\Sigma, X, \tau, x_0, X_f)$: the temporal diagnoser of a DES $\mathcal{X}$, and O: a temporal observation of $\mathcal{X}$
3: **output** $\mathcal{R}$: a regular expression whose language is the temporal diagnosis $\Delta(O)$ (cf. Definition 2)
4: $\mathbb{C} \leftarrow \{(x_0, \varepsilon)\}$
5: **for all** observation $o \in O$ **do**
6: $\mathbb{C}_{\text{new}} \leftarrow \emptyset$
7: **for all** $(x', r') \in \mathbb{C}$ **do**
8: **for all** arc $\langle (x', x), (o, r, f), (x'_2, x_2) \rangle$ in τ **do**
9: $r_2 \leftarrow r' r f$
10: **if** $(x'_2, r'_2) \in \mathbb{C}_{\text{new}}$ **then**
11: Substitute $(x'_2, (r'_2|r_2))$ for (x'_2, r'_2) in $\mathbb{C}_{\text{new}}$
12: **else**
13: Insert (x'_2, r_2) into $\mathbb{C}_{\text{new}}$
14: **end if**
15: **end for**
16: **end for**
17: $\mathbb{C} \leftarrow \mathbb{C}_{\text{new}}$
18: **end for**
19: Remove from $\mathbb{C}$ every context (x, r) where x does not include any final state
20: **if** $\mathbb{C} = \{(x, r)\}$ **then**
21: $\mathcal{R} \leftarrow r\mathcal{L}(x)$
22: **else if** $\mathbb{C} = \{(x_1, r_1), \ldots, (x_k, r_k)\}$ where $k > 1$ **then**
23: $\mathcal{R} \leftarrow (r_1(\mathcal{L}(x_1)))| \ldots |(r_k(\mathcal{L}(x_k)))$
24: **end if**
25: **end procedure**

5. Ideal Diagnosis

A temporal diagnoser $\mathcal{X}^\Delta$ is compiled knowledge built offline that allows for the efficient online generation of a temporal diagnosis $\Delta(O)$ of $\mathcal{X}$ by means of an algorithm called *Ideal Diagnosis*. The "ideal" qualifier indicates that the actual generation of the whole temporal diagnoser is prohibitive in real applications, owing to the exponential explosion of the number of states. Roughly, $\mathcal{X}^\Delta$ is traversed based on O and the regular expressions marking the transitions of $\mathcal{X}^\Delta$ are concatenated in the given order. When the transition relevant to the last observation in O is traversed and a final state x_f is entered, the regular expression composed so far is eventually appended with the diagnosis language $\mathcal{L}(x_f)$, which was itself precomputed offline. Since $\mathcal{X}^\Delta$ is an NFA, several paths can generate the same temporal observation O; therefore, the final regular expression is in general composed by the alternative of several subexpressions. The pseudocode of *Ideal Diagnosis* is listed in Algorithm 1 (lines 1–25). It takes as input a temporal diagnoser $\mathcal{X}^\Delta$ and a temporal observation O, and generates as output a regular expression $\mathcal{R}$ whose language equals $\Delta(O)$. To this end, the algorithm exploits a set of *contexts*, namely $\mathbb{C}$, each context being a pair (x, r), where x is a state of $\mathcal{X}^\Delta$ and r a regular expression on the faults of $\mathcal{X}$. Initially, $\mathbb{C}$ includes just the initial context (x_0, ε), where x_0 is the initial state of $\mathcal{X}^\Delta$ (line 4). Then, a loop is performed on the observations in O (lines 5–18). At each iteration, a new set of contexts, namely $\mathbb{C}_{\text{new}}$ is generated based on the current content of $\mathbb{C}$. Specifically, for each context (x', r') in $\mathbb{C}$ and for each arc of $\mathcal{X}^\Delta$ exiting x' and marked with the triple (o, r, f), where o is the current observation, a regular expression $r_2 = r' r f$ is computed (line 9). In fact, r' accounts for the faults up to x', r accounts for the faults up to the internal state x of x', and f is the (possibly empty) fault associated with the component transition that is observable by means of o. The update of $\mathbb{C}_{\text{new}}$ is performed in lines 10–14, depending on whether a context involving the reached state x'_2 exists in $\mathbb{C}_{\text{new}}$ or not. If a context (x'_2, r'_2) exists, then its regular expression is extended with the alternative r_2, thereby yielding the updated context $(x'_2, (r'_2|r_2))$ (line 11). Otherwise, a new context (x'_2, r_2) is created (line 13). Before the end of the iteration, $\mathbb{C}$ is replaced with $\mathbb{C}_{\text{new}}$ (line 17). When all the observations have been considered

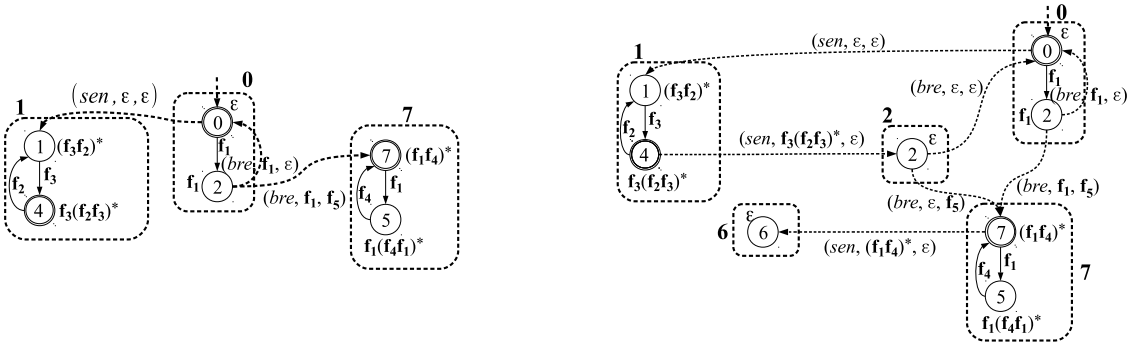


Fig. 4. Partial temporal diagnoser $\mathcal{P}^{\delta}$, prefix of $\mathcal{P}^{\Delta}$ (left), and $\mathcal{P}^{\delta}$ upgraded based on the observation pattern $O^*_{\mathcal{S}}$ shown on the right of Figure 5.

(termination of the outer loop), every context (x, r) in $\mathbb{C}$ that does not include any final state is removed (line 19), as there is no trajectory ending in x. Eventually, the regular expression $\mathcal{R}$ is determined (lines 20–24). Two cases are possible for $\mathbb{C}$: it contains either one context (x, r) or $k > 1$ contexts. In the first case (lines 20–21), $\mathcal{R}$ is generated by appending r with the diagnosis language of x, thereby obtaining $\mathcal{R} = r\mathcal{L}(x)$. This is because r accounts for the faults up to the initial state of x, while $\mathcal{L}(x)$ accounts for the faults within x (up to any final state within x). In the second case (lines 22–23), since several contexts exist, the same operation is performed for each context (x_i, r_i), $i \in [1\,..\,k]$, thereby yielding the regular expression $\mathcal{R}$ that is composed of the alternatives $r_i(\mathcal{L}(x_i))$.

Example 5. With reference to the temporal diagnoser $\mathcal{P}^{\Delta}$ displayed in Figure 3, let $O = [bre, sen]$ be the temporal observation of DES $\mathcal{P}$. Based on line 4 of Algorithm 1, we have $\mathbb{C} = \{(0, \varepsilon)\}$. On the first observation, namely *bre*, two arcs are involved in the loop (line 8), namely $\langle (0,2), (bre, \mathbf{f_1}, \varepsilon), (0,0)\rangle$ and $\langle (0,2), (bre, \mathbf{f_1}, \mathbf{f_5}), (7,7)\rangle$. With the first arc, we have $r_2 = \mathbf{f_1}$ and, with the second arc, $r_2 = \mathbf{f_1 f_5}$. Thus, $\mathbb{C}_{\text{new}} = \{(0, \mathbf{f_1}), (7, \mathbf{f_1 f_5})\}$ (line 13). On the observation *sen* (second iteration of the outer loop), we have $\mathbb{C} = \{(0, \mathbf{f_1}), (7, \mathbf{f_1 f_5})$. Now, the arcs involved in line 8 are $\langle (0,0), (sen, \varepsilon, \varepsilon), (1,1)\rangle$ and $\langle (7,7), (sen, (\mathbf{f_1 f_4})^*, \varepsilon), (6,6)\rangle$. With the first arc, we have $r_2 = \mathbf{f_1}$ and, with the second arc, $r_2 = \mathbf{f_1 f_5 (f_1 f_4)^*}$. Thus, $\mathbb{C}_{\text{new}} = \{(1, \mathbf{f_1}), (6, \mathbf{f_1 f_5 (f_1 f_4)^*})\}$. Since there is no further observation, the loop terminates (line 18). In line 19, the context $((6, \mathbf{f_1 f_5 (f_1 f_4)^*})$ is removed from $\mathbb{C}$ because there is no final state in 6. Eventually, since $\mathbb{C}$ includes just one context, $\mathcal{R}$ is computed in line 21, namely $\mathcal{R} = \mathbf{f_1}\mathcal{L}(1) = \mathbf{f_1 f_3 (f_2 f_3)^*}$. Remarkably, the language of $\mathcal{R}$ equals the temporal diagnosis $\Delta(O)$ that was determined in Example 2 by inspection of the space $\mathcal{P}^*$.

6. Practical Diagnosis

In real applications, assuming that a temporal diagnoser is available in its entirety is impractical, even if the generation of the diagnoser is performed offline, because of the exponential explosion of the set of states involved. Hence, we propose a viable approach called *practical diagnosis*, in which a *partial temporal diagnoser* is generated upfront and subsequently extended either offline, based on meaningful behavioral scenarios, or when being operated online. A similar consideration applies to the space $\mathcal{X}^*$, whose construction is assumed to be impractical. Hence, hereafter, a notation like $\langle x, t, x'\rangle \in \mathcal{X}^*$ does not assume that $\mathcal{X}^*$ is available: it is only a shorthand for stating that the component transition t is triggerable at the state x of $\mathcal{X}$.

Definition 5. *Let $\mathcal{X}^{\Delta}$ be the temporal diagnoser of $\mathcal{X}$. A* partial temporal diagnoser *of $\mathcal{X}$, denoted $\mathcal{X}^{\delta}$, is a connected subgraph of $\mathcal{X}^{\Delta}$ that includes the initial state of $\mathcal{X}^{\Delta}$.*

Example 6. With reference to the temporal diagnoser $\mathcal{P}^{\Delta}$ displayed in Figure 3, a partial temporal diagnoser $\mathcal{P}^{\delta}$ is shown on the left of Figure 4. In particular, $\mathcal{P}^{\delta}$ is a *prefix* of $\mathcal{P}^{\Delta}$ at distance 1, that is, the subgraph of $\mathcal{P}^{\Delta}$ that includes all the states and transitions that can be reached (from the initial state) by one (dashed) transition.

Once an initial partial temporal diagnoser $\mathcal{X}^{\delta}$ has been somehow generated, for instance a prefix of the (whole) temporal diagnoser $\mathcal{X}^{\Delta}$, it can be *upgraded* in several ways, such as by integrating into it the knowledge about the evolutions concisely represented by some *behavioral scenarios* [5, 3, 4].

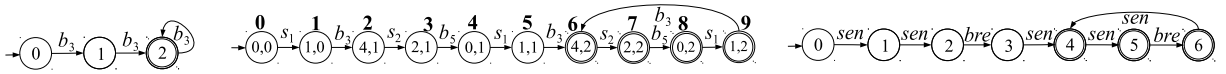


Fig. 5. DFA of scenario $\mathcal{S}$ (left), abduction $P^*_{\mathcal{S}}$ (center), and observation pattern $O^*_{\mathcal{S}}$ (right).

Definition 6. *Let $\mathcal{X}$ be a DES and $\mathbb{T}$ a subset of the component transitions in $\mathcal{X}$. A* behavioral scenario *of $\mathcal{X}$ is a pair $\mathcal{S} = (\mathbb{T}, \mathcal{L})$, where $\mathcal{L}$ is a regular language on $\mathbb{T}$.*

Example 7. A scenario where the only malfunction is the breaker being stuck closed can be defined as $\mathcal{S} = (\mathbb{T}, \mathcal{L})$, where $\mathbb{T} = \{s_3, s_4, b_1, b_2, b_3, b_4, b_7, v_8\}$ and $\mathcal{L} = b_3 b_3^+$ (repetition of b_3 at least twice).

To upgrade a partial temporal diagnoser $\mathcal{X}^\delta$ so that it embeds the set of temporal observations generated by a scenario $\mathcal{S}$, we need to synchronize $\mathcal{S}$ with the behavior of $\mathcal{X}$, as described below.

Definition 7. *Let $\mathcal{S} = (\mathbb{T}, \mathcal{L})$ be a scenario of $\mathcal{X}$. The* restriction *of a trajectory T in $\mathcal{X}^*$ on $\mathbb{T}$ is the sequence $T_{\mathbb{T}} = [\, t \mid t \in T, t \in \mathbb{T}\,]$. The* abduction *of $\mathcal{S}$, $\mathcal{X}^*_{\mathcal{S}}$, is a DFA whose language is the set $\{\, T \mid T \in \mathcal{X}^*, T_{\mathbb{T}} \in \mathcal{L}\,\}$.*

In other words, the abduction of a scenario $\mathcal{S}$ is a subspace of $\mathcal{X}^*$ where each trajectory T conforms with a string of the scenario, in the sense that the subsequence of the component transitions in T that are in $\mathbb{T}$ is a string in $\mathcal{L}$.

Example 8. With reference to the behavioral scenario $\mathcal{S}$ defined in Example 7, shown in Figure 5 are the DFA recognizing $\mathcal{S}$ (left) and the abduction $\mathcal{P}^*_{\mathcal{S}}$ (center). Each state of $\mathcal{P}^*_{\mathcal{S}}$ is a pair (p, d), where p is a state of $\mathcal{P}^*$ and d is a state of the DFA. A state is final when d is final ($d = 2$).

The next step is to distill the observation language of the abduction, namely the set of temporal observations generated by the trajectories in the abduction (the *observation pattern*).

Definition 8. *Let $\mathcal{X}$ be a DES and* **O** *the domain of observations involved in the mapping table $\mu(\mathcal{X})$. An* observation pattern *O^* of $\mathcal{X}$ is a DFA whose language is a set of strings on* **O**.

The definition of an observation pattern is general in nature. Still, meaningful observation patterns can be derived from abductions. Specifically, each symbol t marking a transition $\langle a, t, a' \rangle$ in an abduction $\mathcal{X}^*_{\mathcal{S}}$ is replaced with a (possibly ε) observation o, where $(t, o, f) \in \mu(\mathcal{X})$. The resulting NFA is then determinized into an equivalent (possibly minimized) DFA, which is by definition the observation pattern of the scenario $\mathcal{S}$, denoted $O^*_{\mathcal{S}}$. Remarkably, the language of $O^*_{\mathcal{S}}$ is the set of temporal observations associated with the set of trajectories in the abduction, with each trajectory being a mode in which the scenario $\mathcal{S}$ manifests itself in $\mathcal{X}^*$.

Example 9. With reference to the scenario $\mathcal{S}$ defined in Example 7 and the abduction $\mathcal{P}^*_{\mathcal{S}}$ displayed in the center of Figure 5, shown on the right of Figure 5 is the observation pattern $O^*_{\mathcal{S}}$.

To upgrade a partial temporal diagnoser $\mathcal{X}^\delta$ based on an observation pattern O^*, Algorithm 2 is used (lines 1–28). Each state x^δ in $\mathcal{X}^\delta$ is assumed to be marked with a *labeling set* (initially empty), denoted $\Lambda(x^\delta)$, which contains states of O^*. This serves to synchronize $\mathcal{X}^\delta$ with O^* avoiding duplications of $\mathcal{X}^\delta$ states, as well as endless loops caused by cycles in O^*. A unmarked state ω in $\Lambda(x^\delta)$ means that the transitions exiting ω in O^* need to be synchronized with the transitions exiting x^δ in $\mathcal{X}^\delta$. If a transition is missing in $\mathcal{X}^\delta$, it is created, possibly along with its target state, a fault space, which is marked with the relevant labeling set (line 18). Once all transitions exiting ω in O^* have been processed, ω is marked in $\Lambda(x^\delta)$ (line 24). The run terminates when there is no unmarked ω in any labeling set.

Example 10. Let $\mathcal{P}^\delta$ be the partial temporal diagnoser defined in Example 6 and shown on the left of Figure 4. Let $O^*_{\mathcal{S}}$ be the observation pattern displayed on the right of Figure 5. The partial explainer resulting from the application of Algorithm 2 on $\mathcal{P}^\delta$ and $O^*_{\mathcal{S}}$ is displayed on the right of Figure 4.

Assuming a partial diagnoser, the *Ideal Diagnosis* specified in Algorithm 1 needs to be revised. Specifically, when considering a new observation $o \in O$ in line 5, we have to check whether the transition function of the relevant state in the partial temporal diagnoser needs to be extended by o. If so, the partial temporal diagnoser is upgraded in a way that is similar to the mode in which Algorithm 2 works. As such, this new algorithm, called *Practical Diagnosis*, not only generates the temporal diagnosis $\Delta(O)$, but possibly upgrades the partial temporal diagnoser when required.

Algorithm 2 *Diagnoser Upgrade*

procedure DIAGNOSER UPGRADE($\mathcal{X}^\delta$, O^*)
  input $\mathcal{X}^\delta$: a partial temporal diagnoser of $\mathcal{X}$, and O^*: an observation pattern for $\mathcal{X}$, with initial state ω_o
  side effects: the partial temporal diagnoser $\mathcal{X}^\delta$ is upgraded based on O^*
  Insert ω_0 into the (initially empty) labeling set $\Lambda(x_0^\delta)$, where x_0^δ is the initial state of $\mathcal{X}^\delta$
  repeat
    Let $\Lambda(x^\delta)$ be a set including an unmarked pattern state ω
    for all unmarked pattern state $\omega \in \Lambda(x^\delta)$ **do**
      for all transition $\langle \omega, o, \omega' \rangle$ in O^* **do**
        if there is a transition exiting x^δ marked with $(o, \mathcal{L}, f)$ **then**
          for all transition $\langle x^\delta, (o, \mathcal{L}, f), x'^\delta \rangle$ in $\mathcal{X}^\delta$ **do**
            Insert ω' into $\Lambda(x'^\delta)$, unless ω' is included in $\Lambda(x'^\delta)$ already
          end for
        else
          for all $x \in x^\delta, \langle x, t, x' \rangle \in \mathcal{X}^*, (t, o, f) \in \mu(\mathcal{X})$ **do**
            Let x'^δ denote the fault space of x', namely $\mathcal{X}^*_{x'}$
            if $\mathcal{X}^\delta$ does not include the state x'^δ **then**
              Create the state $x'^\delta = \mathcal{X}^*_{x'}$ in $\mathcal{X}^\delta$
              Mark x'^δ with the (singleton) labeling set $\{\omega'\}$
            end if
            Create the transition $\langle x^\delta, (o, \mathcal{L}(x), f), x'^\delta \rangle$ in $\mathcal{X}^\delta$
          end for
        end if
      end for
      Mark the pattern state ω within the labeling set $\Lambda(x^\delta)$
    end for
  until there is no labeling set Λ including an unmarked state
  Empty all the nonempty labeling sets Λ in $\mathcal{X}^\delta$
end procedure

7. Conclusion

A shift from a set-oriented to a temporal-oriented perspective in diagnosis of DESs has been proposed in this paper. The motivation for this shift is grounded on critical-decision making, where the temporal information embedded in candidates may be essential for a crystal clear explanation of the temporal observation. This temporal-oriented paradigm is novel inasmuch all approaches to diagnosis of DESs in the literature, including the seminal diagnoser approach [20] (along with all its variants), are set-oriented. In contrast with a set-oriented setting, where a candidate is a finite *set* of faults, with neither temporal ordering nor duplicates, in our temporal-oriented perspective a candidate is a (possibly unbounded) *sequence* of faults, with reciprocal temporal ordering and multiple occurrences of the same fault being manifested. The supposedly unacceptable burden of unbounded candidates and/or infinite sets of candidates is dominated by a notation based on regular expressions on faults.

Fault detection in DESs has been generalized in [11] to the recognition of a *supervision pattern*, this being a DFA that can represent the ordered occurrences of (possibly multiple) faults. Thus, it is tempting to believe that diagnosis with supervision patterns somewhat resembles the approach proposed in this paper. Still, each supervision pattern requires the specification of a finite automaton whose regular language is a set of strings of transitions. By contrast, the approach in this paper is not given any automaton upfront recognizing a language; instead, it generates a regular expression representing the language of the faults of all the trajectories that imply the temporal observation. Moreover, the output of the supervision pattern approach only clarifies whether the pattern has occurred. In doing so, however, it does not compute the number of its occurrences, nor does it show the reciprocal order of these occurrences and those of individual faults within the trajectories implying the temporal observation. In the view of the current paper,

instead, if a fault is associated with a pattern, this can be part of a temporal fault as all other faults are. In other words, supervision patterns, as well as other "complex" faults [13, 16], can be embedded in temporal faults and treated homogeneously.

Future research includes an extensive experimental activity based on synthetic benchmarks as well as the adoption of temporal diagnosis for other classes of DESs, including complex DESs [12, 14]. Applying the approach described in this paper to a real-world case study is also a challenge for the future. Several applications domains could be targeted besides the one that inspired the running example in this paper, namely protection apparata for power transmission networks, including home automation and pandemic monitoring.

References

[1] Baroni, P., Lamperti, G., Pogliano, P., Zanella, M., 1999. Diagnosis of large active systems. Artificial Intelligence 110, 135–183. doi:10.1016/S0004-3702(99)00019-3.

[2] Basile, F., 2014. Overview of fault diagnosis methods based on Petri net models, in: Proceedings of the 2014 European Control Conference, ECC 2014, pp. 2636–2642. doi:10.1109/ECC.2014.6862631.

[3] Bertoglio, N., Lamperti, G., Zanella, M., 2019a. A posteriori diagnosis of discrete-event systems with symptom dictionary and scenarios, in: Wotawa, F., Friedrich, G., Pill, I., Koitz-Hristov, R., Ali, M. (Eds.), Advances and Trends in Artificial Intelligence. From Theory to Practice. IEA/AIE 2019. Springer International Publishing, Cham. volume 11606 of *Lecture Notes in Computer Science*, pp. 325–333. doi:10.1007/978-3-030-22999-3_29.

[4] Bertoglio, N., Lamperti, G., Zanella, M., 2019b. Temporal diagnosis of discrete-event systems with dual knowledge compilation, in: Holzinger, A., Kieseberg, P., Weippl, E., Tjoa, A.M. (Eds.), Machine Learning and Knowledge Extraction. Springer, Berlin. volume 11713 of *Lecture Notes in Computer Science*, pp. 333–352. doi:10.1007/978-3-030-29726-8_21.

[5] Bertoglio, N., Lamperti, G., Zanella, M., 2020. Intelligent diagnosis of discrete-event systems with preprocessing of critical scenarios, in: Czarnowski, I., Howlett, R., Jain, L. (Eds.), Intelligent Decision Technologies 2019. Springer, Singapore. volume 142 of *Smart Innovation, Systems and Technologies*, pp. 109–121. doi:10.1007/978-981-13-8311-3_10.

[6] Brand, D., Zafiropulo, P., 1983. On communicating finite-state machines. Journal of the ACM 30, 323–342. doi:10.1145/322374.322380.

[7] Brzozowski, J., McCluskey, E., 1963. Signal flow graph techniques for sequential circuit state diagrams. IEEE Transactions on Electronic Computers EC-12, 67–76.

[8] Cassandras, C., Lafortune, S., 2008. Introduction to Discrete Event Systems. second ed., Springer, New York.

[9] Cong, X., Fanti, M., Mangini, A., Li, Z., 2018. Decentralized diagnosis by Petri nets and integer linear programming. IEEE Transactions on Systems, Man, and Cybernetics: Systems 48, 1689–1700.

[10] Hamscher, W., Console, L., de Kleer, J. (Eds.), 1992. Readings in Model-Based Diagnosis. Morgan Kaufmann, San Mateo, CA.

[11] Jéron, T., Marchand, H., Pinchinat, S., Cordier, M., 2006. Supervision patterns in discrete event systems diagnosis, in: Workshop on Discrete Event Systems (WODES 2006), IEEE Computer Society, Ann Arbor, MI. pp. 262–268.

[12] Lamperti, G., Quarenghi, G., 2016. Intelligent monitoring of complex discrete-event systems, in: Czarnowski, I., Caballero, A., Howlett, R., Jain, L. (Eds.), Intelligent Decision Technologies 2016. Springer International Publishing Switzerland. volume 56 of *Smart Innovation, Systems and Technologies*, pp. 215–229. doi:10.1007/978-3-319-39630-9_18.

[13] Lamperti, G., Zanella, M., 2011. Context-sensitive diagnosis of discrete-event systems, in: Walsh, T. (Ed.), Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), AAAI Press, Barcelona, Spain. pp. 969–975.

[14] Lamperti, G., Zanella, M., Zhao, X., 2018a. Abductive diagnosis of complex active systems with compiled knowledge, in: Thielscher, M., Toni, F., Wolter, F. (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference (KR2018), AAAI Press, Tempe, Arizona. pp. 464–473.

[15] Lamperti, G., Zanella, M., Zhao, X., 2018b. Introduction to Diagnosis of Active Systems. Springer, Cham. doi:10.1007/978-3-319-92733-6.

[16] Lamperti, G., Zhao, X., 2014. Diagnosis of active systems by semantic patterns. IEEE Transactions on Systems, Man, and Cybernetics: Systems 44, 1028–1043. doi:10.1109/TSMC.2013.2296277.

[17] Lunze, J., 2000. Diagnosis of quantized systems based on a timed discrete-event model. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 30, 322–335.

[18] Pencolé, Y., Steinbauer, G., Mühlbacher, C., Travé-Massuyès, L., 2017. Diagnosing discrete event systems using nominal models only, in: 28th International Workshop on Principles of Diagnosis (DX 2017), Brescia, Italy. pp. 169–183.

[19] Reiter, R., 1987. A theory of diagnosis from first principles. Artificial Intelligence 32, 57–95.

[20] Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D., 1995. Diagnosability of discrete-event systems. IEEE Transactions on Automatic Control 40, 1555–1575.

[21] Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D., 1996. Failure diagnosis using discrete-event models. IEEE Transactions on Control Systems Technology 4, 105–124.

[22] Struss, P., 1997. Fundamentals of model-based diagnosis of dynamic systems, in: Fifteenth International Joint Conference on Artificial Intelligence (IJCAI 1997), Nagoya, Japan. pp. 480–485.