Computing and Informatics, Vol. 41, 2022, 116–134, doi: 10.31577/cai_2022_1_116

FAULT DIAGNOSIS OF DISCRETE-EVENT SYSTEMS FROM ABSTRACT OBSERVATIONS

Gianfranco Lamperti, Marina Zanella

Department of Information Engineering
University of Brescia
Via Branze 38
25123 Brescia, Italy
e-mail: {gianfranco.lamperti, marina.zanella}@unibs.it

Xiangfu Zhao

School of Computer and Control Engineering
Yantai University
30, Qingquan RD, Laishan District
Yantai 264005, China
e-mail: xiangfuzhao@gmail.com

Abstract. Active systems (ASs) are a special class of (asynchronous) discrete-event systems (DESs). An AS is represented by a network of components, where each component is modeled as a communicating automaton. Diagnosing a DES amounts to finding out possible *faults* based on the DES model and a sequence of observations gathered while the DES is being operated. This is why the diagnosis engine needs to know what is observable in the behavior of the DES and what is not. The notion of *observability* serves this purpose. In the literature, defining the observability of a DES boils down to qualifying the state transitions of components either as observable or unobservable, where each observable transition manifests itself as an *observation*. Still, looking at the way humans observe reality, typically by associating a collection of events with a single, abstract perception, the state-of-the-art notion of DES observability appears somewhat narrow. This paper presents, a generalized notion of observability, where an observation is abstract rather than concrete, since it is associated with a DES behavioral scenario rather than a single component transition. To support the online diagnosis engine, knowledge compila-

tion is performed offline. The outcome is a set of data structures, called *watchers*, which allow for the tracking of abstract observations.

Keywords: Model-based diagnosis, abduction, active systems, discrete-event systems, finite automata, observability, abstract observations, uncertainty

1 INTRODUCTION

Model-based diagnosis [29, 14] exploits the model of a system in order to find the causes of its abnormal behavior, based on some observations. For diagnosing a dynamical system [32], a discrete-event system (DES) model can be adopted [9], this being either a Petri net [16, 8, 2, 10] or a net of communicating finite automata, one automaton for each component [1, 11, 27, 12, 17, 18, 13, 24]. Although an automaton relevant to a DES component can represent just its nominal behavior [28], usually each state transition is qualified as either *normal* or *faulty*, as in the seminal work by Sampath et al. [30, 31].

The input of the DES diagnosis task is a sequence of temporally ordered observations, called a *temporal observation*. The output is a set of *candidates*, each candidate being a set of *faults*, where a fault is associated with a faulty transition.

In recent years, in contrast with this general set-oriented approach to diagnosis of DESs, a *temporal-oriented* approach to diagnosis has been proposed [5, 3, 4], where a candidate is a (possibly unbounded) sequence of faults. Diagnosing a DES becomes a sort of *abductive* reasoning, inasmuch as the set of candidates is based on the (possibly infinite) set of *trajectories* (sequences of component transitions) of the DES that entail the temporal observation. Since the domain of faults is finite, both the candidates and the diagnosis output are finite and bounded.

The process of abstracting abnormality and observability of a DES from the behavioral models of its components started long ago. In the approach proposed in [21], these properties are represented separately from the behavioral models. Thus, there may be several DESs sharing the same network of components and differing in their abnormality and/or observability. This is reasonable, since the abnormality of a DES not only depends on its component transitions, but on the context where the DES is used as well as on the events the diagnostician is interested to track. Similarly, the observability of a DES does not depend on its component transitions only, but on the operating context, as different contexts may cause different sets of observations, as well as on the sensing apparatus. The sensing equipment depends in turn on the events the stakeholders (including the diagnostician) are interested to perceive.

Abnormality in DESs was further generalized to a *pattern* in [15], namely a deterministic finite automaton (DFA) that can represent specific combinations of faults, which can be detected by the diagnosis engine based on the temporal observation of the DES. Instead of a single pattern, several patterns can actually be consid-

ered, and a distinct generalized fault can be associated with each of them, like in [22, 26], where generalized faulty events are part of the candidates. Inspired by the generalization of faulty events, this paper generalizes the notion of DES observability.

In the literature, observability is expressed by a surjective function from a subset of the component transitions (domain) to a set of observations (codomain). This paper defines observability as a function from a regular language over component transitions to a set of observations. Hence, what is generalized is solely the *domain* of the function. Still, observations now become *abstract*, since they somehow symbolize fragments of the DES behavior.

The generalization of DES observability allows for modeling real-world scenarios where, outside the DES, one can figure out the occurrence of a specific evolution of the DES, rather than a single component transition. After all, this view resembles the way humans observe reality, typically by associating a combination of events with a single, abstract perception. Still, the proposed generalization does not come without difficulties. On the one hand, since the strings of transitions in the domain of the observability function may overlap while the DES is being operated, simple recognizers of regular languages are not sufficient. This is why extended recognizers are generated offline, called the *watchers*. On the other hand, since several strings of transitions may end at the same point of the DES trajectory, several (abstract) observations may be simultaneous. Thus, in general, a trajectory of the DES implies a sequence $\mathbb{O}$ of *sets* of observations.

The temporal observation $\mathcal{O}$ of the DES that is actually perceived by an observer when the DES follows its trajectory is a sequence of observations obtained from $\mathbb{O}$ by substituting each set O of observations with a sequence (one out of the possible permutations of O). Since this set-to-sequence transformation is unpredictable, the diagnosis engine is required to cope with all the possibilities. Based on a temporal observation $\mathcal{O}$, the diagnosis engine is expected to single out the whole (possibly infinite) set of trajectories entailing $\mathcal{O}$ in order to eventually generate the (finite) set of candidates.

2 DISCRETE-EVENT SYSTEM

A DES is a network of components connected by links. A component is endowed with a set of input pins and a set of output pins. A link connects an output pin of a component with an input pin of another component, where each pin is entered/exited by a single link. Each component is modeled by a communicating automaton [7], where a transition is triggered by an event either occurring in the external world or being ready at an input pin of the component. The occurrence of a transition consumes the triggering event and possibly generates new events on some output pins, thereby providing triggering events to other components. This results in a reaction of the DES, where a series of component transitions move the DES from the initial state to a final state, with all events being consumed.

Figure 1. Thermovalve $\mathcal{V}$ (center) and models of the sensor s (left) and the valve v (right)

Component transition	*Description*
$v_1 = \langle closed, (op, \emptyset), open \rangle$	v reacts to the open event by opening
$v_2 = \langle open, (cl, \emptyset), closed \rangle$	v reacts to the close event by closing
$v_3 = \langle closed, (op, \emptyset), closed \rangle$	v does not react to the open event and remains closed
$v_4 = \langle open, (cl, \emptyset), open \rangle$	v does not react to the close event and remains open
$v_5 = \langle closed, (cl, \emptyset), closed \rangle$	v reacts to the close event by remaining closed
$v_6 = \langle open, (op, \emptyset), open \rangle$	v reacts to the open event by remaining open
$v_7 = \langle closed, (cl, \emptyset), open \rangle$	v reacts to the close event by opening
$v_8 = \langle open, (op, \emptyset), closed \rangle$	v reacts to the open event by closing
$s_1 = \langle norm, (ko, \{op\}), high \rangle$	s detects high temperature and generates the open event
$s_2 = \langle high, (ok, \{cl\}), norm \rangle$	s detects low temperature and generates the close event
$s_3 = \langle norm, (ko, \{cl\}, norm \rangle$	s detects high temperature, yet generates the close event
$s_4 = \langle high, (ok, \{op\}), high \rangle$	s detects low temperature, yet generates the open event

Table 1. Description of component transitions

Example 1. Represented in the center of Figure 1 is a DES, named $\mathcal{V}$, modeling a thermovalve that is composed of a temperature sensor s and a valve v. A link connects the (unique) output pin of s with the (unique) input pin of v. In normal behavior, when the temperature exceeds a given threshold (external event ko), the sensor commands the valve to open. Afterwards, if the temperature becomes normal again (external event ok), the sensor commands the valve to close. The communicating automata of s and v are displayed on the left and right side of the figure, respectively. The details inherent to the component transitions are listed in Table 1. Each component transition is denoted as a triple $\langle x, (e, E), x' \rangle$, where x is the current state of the component, e is the triggering (input) event, E is the set of output events, and x' is the new component state. As defined in the table, both the sensor and the valve may misbehave by performing faulty transitions. For instance, the sensor may command the valve to close rather than to open (transition s_3); on its part, the valve, when expected to open, may remain closed (transition v_3).

The behavior of a DES is constrained by its topology and the models of the components involved. These constraints confine the behavior within a finite space, called *behavior space*.

Definition 1 (Behavior Space)**.** Let $\mathcal{X}$ be a DES. The *behavior space* of $\mathcal{X}$ is a DFA[1]

$$\mathcal{X}^* = (\Sigma, X, \tau, x_0, X_\mathrm{f}) \tag{1}$$

where Σ is the alphabet, comprising the set of component transitions, X is the set of states (S, E), where S is a tuple of component states and E is a tuple of (possibly empty) events that are ready inside the links, $x_0 = (S_0, E_0)$ is the initial state, where all events in E_0 are empty, $X_\mathrm{f} \subseteq X$ is the set of final states $(S_\mathrm{f}, E_\mathrm{f})$ such that all events in E_f are empty, $\tau : X \times \Sigma \mapsto X$ is the transition function, where $\tau(x, t) = x'$ when t is triggerable at state x and x' is the state reached by the consumption of the input event and the generation of the output events relevant to t.

Definition 2 (Trajectory)**.** A sequence of component transitions in the language of a behavior space $\mathcal{X}^*$, namely $T = [t_1, \ldots, t_n]$, is a *trajectory* of $\mathcal{X}$. A prefix of T is a *semi-trajectory* of $\mathcal{X}$. Let $\mathbb{T}$ be a set of component transitions in $\mathcal{X}$. The *restriction* of T on $\mathbb{T}$ is $T_{[\mathbb{T}]} = [t \mid t \in T, t \in \mathbb{T}]$.

Example 2. Shown in Figure 2 is the behavior space $\mathcal{V}^*$ of the thermovalve $\mathcal{V}$, where each state is a triple: a state of the sensor s, a state of the valve v, and an event that is ready at the input pin of v (ϵ denotes no event). States are renamed $0, \ldots, 7$, where 0, 3, 4, and 7 are final (ϵ indicates that the link is empty). The space involves an infinite number of trajectories, one of them being $T = [s_1, v_1, s_2, v_4, s_3, v_4]$.

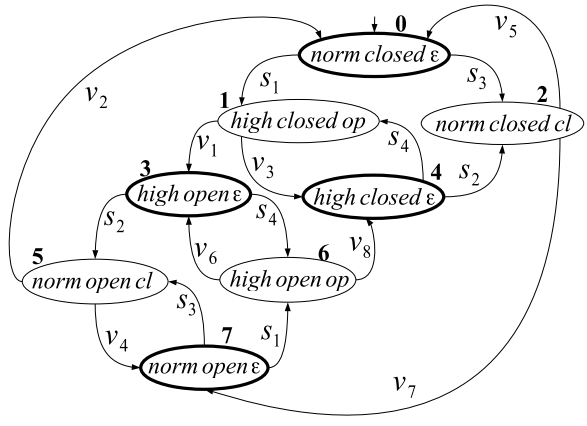

Figure 2. Behavior space $\mathcal{V}^*$, where final states are highlighted in bold

[1] In general, a finite automaton, be it a DFA or an NFA (nondeterministic finite automaton), is defined as a 5-tuple $(\Sigma, S, \tau, s_0, S_\mathrm{f})$, where Σ is the alphabet, S is the set of states, τ is the transition function, s_0 is the initial state, and S_f is the set of final states.

3 DIAGNOSIS PROBLEM

In order to perform the diagnosis task, the specification of a DES needs to be extended with information indicating which behavior is normal and which is abnormal. In our approach, abnormality is associated with faulty transitions.

Definition 3 (Abnormality)**.** Let $\mathbf{T}$ be the domain of component transitions of a DES $\mathcal{X}$, and let $\mathbf{F}$ be a domain of symbols called *faults*. The *abnormality* of $\mathcal{X}$ is a set of associations between component transitions and faults, namely $Abn(\mathcal{X}) \subseteq \mathbf{T} \times \mathbf{F}$, where each transition appears at most once. If $(t, f) \in Abn(\mathcal{X})$, then t is *faulty*, else t is *normal*.

Each trajectory is associated with a *diagnosis*.

Definition 4 (Diagnosis)**.** Let $T = [t_1, \ldots, t_n]$ be a trajectory of a DES $\mathcal{X}$. The *diagnosis* δ of T is the set of faults associated with the faulty transitions in T, namely

$$\delta(T) = \{f \mid t \in T, (t, f) \in Abn(\mathcal{X})\}. \tag{2}$$

Since a diagnosis is a set, possible repetitions of the same fault are missing.

Example 3. Outlined in Table 2 is the description of the abnormality $Abn(\mathcal{V})$ for the thermovalve. Considering the trajectory $T = [s_1, v_1, s_2, v_4, s_3, v_4]$ of $\mathcal{V}$ (cf. Example 2), based on Definition 4, we have $\delta(T) = \{vro, ssc\}$, indicating that the valve remains open upon receiving the close command from the sensor (vro), and the sensor commands the valve to close rather than to open (ssc).

t	f	*Fault Description*
s_3	*ssc*	The sensor commands the valve to close rather than to open
s_4	*sso*	The sensor commands the valve to open rather than to close
v_3	*vrc*	The valve remains closed upon receiving the open command
v_4	*vro*	The valve remains open upon receiving the close command
v_7	*vop*	The valve opens upon receiving the close command
v_8	*vcl*	The valve closes upon receiving the open command

Table 2. Abnormality $Abn(\mathcal{V})$ of the thermovalve

The behavior of the DES is observable in that an observation is associated with a regular language that is defined by a regular expression on a set of component transitions.

Definition 5 (Observability)**.** Let $\mathbf{T}$ be the domain of component transitions of a DES $\mathcal{X}$, let $\mathbf{L}$ be the domain of regular languages on subsets of $\mathbf{T}$, and let $\mathbf{O}$ be a domain of symbols called *observations*. The *observability* of $\mathcal{X}$ is a relation

$$Obs(\mathcal{X}) \subseteq 2^{\mathbf{T}} \times \mathbf{L} \times \mathbf{O}. \tag{3}$$

Each element in $Obs(\mathcal{X})$ is a triple $(\mathbb{T}, \mathcal{L}, o)$, where $\mathbb{T}$ is a set of component transitions, $\mathcal{L}$ is a regular language defined by a regular expression on $\mathbb{T}$, and o is the observation.

Example 4. Listed in Table 3 is the observability $Obs(\mathcal{V})$ of the thermovalve. Each row defines an observation o that is associated with a regular language $\mathcal{L}$, where the operators '|' and '+' represent the alternative and the repetition one or more times, respectively. In particular, the language $v_3\, v_3^+ \mid v_4\, v_4^+$ of the the observation *stk* includes the strings of at least two consecutive transitions, either v_3 or v_4, indicating that the valve is either stuck closed or stuck open (cf. Table 2). Likewise, the observation *bal* arises when a sequence $[s_3, v_4]$ occurs, namely when the sensor commands the valve to close rather than to open, with the valve remaining open upon the (faulty) close command. This way, the net effect of the two faulty transitions is null (v_4 balances s_3).

$\mathbb{T}$	$\mathcal{L}$	o	*Observation Description*
$\{s_1, s_2, s_3, s_4\}$	$s_1 \mid s_2 \mid s_3 \mid s_4$	*sns*	The sensor performs a transition
$\{v_1, v_2, v_7, v_8\}$	$v_1 \mid v_2 \mid v_7 \mid v_8$	*vlv*	The valve either opens or closes
$\{v_1, v_2, v_3, v_4, v_7, v_8\}$	$v_3\, v_3^+ \mid v_4\, v_4^+$	*stk*	The valve looks somehow stuck
$\{s_3, v_4\}$	$s_3\, v_4$	*bal*	The faulty valve balances the faulty sensor

Table 3. Observability $Obs(\mathcal{V})$ of the thermovalve

Given a triple $(\mathbb{T}, \mathcal{L}, o) \in Obs(\mathcal{X})$ and a trajectory T of $\mathcal{X}$, the observation o occurs when the restriction of T on $\mathbb{T}$ includes a subsequence that is a string in $\mathcal{L}$. Since several observations may occur at the same time, in theory, T manifests itself as a sequence of sets of observations. However, we assume that observations in the same set are perceived as sequences, where the temporal ordering of each sequence is unpredictable. In other words, a trajectory T of $\mathcal{X}$ is perceived by the observer as a temporal sequence of observations, called a *temporal observation* of $\mathcal{X}$.

Definition 6 (Temporal Observation)**.** Let O be a set of observations. The *space* of O is the set of sequences composed of all the observations in O, $O^* = \{\, [\, o \mid o \in O \,]\, \}$. Let $T = [t_1, \ldots, t_n]$ be a trajectory in $\mathcal{X}^*$ and let $\mathbb{O}$ be a sequence of sets of observations

$$\mathbb{O} = [\, O_i \mid i \in [1\,..\,n], O_i = \{\, o \mid j \in [1\,..\,i], T' = [t_j, \ldots, t_i], \\ (\mathbb{T}, \mathcal{L}, o) \in Obs(\mathcal{X}), T'_{[\mathbb{T}]} \in \mathcal{L} \,\}\,]. \tag{4}$$

Let $\bar{\mathbb{O}}$ be a sequence obtained from $\mathbb{O}$ by replacing each $O_i \in \mathbb{O}$ with a sequence in O_i^*. The concatenation of the sequences in $\bar{\mathbb{O}}$ is a *temporal observation* of $\mathcal{X}$. The whole set of temporal observations relevant to T is the *observation space* of T, denoted T^*. A trajectory T is said to *conform* with a temporal observation $\mathcal{O}$ when $\mathcal{O} \in T^*$.

Example 5. With reference to the behavior space $\mathcal{V}^*$ displayed in Figure 2, consider the trajectory $T = [s_1, v_1, s_2, v_4, s_3, v_4]$ (cf. Example 3). We have $\mathbb{O} = [\{sns\}, \{vlv\}, \{sns\}, \emptyset, \{sns\}, \{bal, stk\}]$ and $T^* = \{[sns, vlv, sns, sns, bal, stk], [sns, vlv, sns, sns, stk, bal]\}$.

Definition 7 (Candidate Set)**.** Let $\mathcal{O}$ be a temporal observation of $\mathcal{X}$. The *candidate set* of $\mathcal{O}$ is a set of diagnoses

$$\Delta(\mathcal{O}) = \{\, \delta(T) \mid T \in \mathcal{X}^*, \mathcal{O} \in T^* \}\,. \tag{5}$$

Solving a diagnosis problem amounts to finding the candidate set of a temporal observation of a DES being operated online.

Example 6. Let $\mathcal{O} = [\, sns, vlv, sns, sns, stk, bal\,]$ be a temporal observation of $\mathcal{V}$. Based on the space $\mathcal{V}^*$ depicted in Figure 2 and the observability $Obs(\mathcal{V})$ defined in Example 4 (cf. Table 3), we have only one trajectory satisfying the conditions in Equation (5), namely $T = [s_1, v_1, s_2, v_4, s_3, v_4]$. Hence, based on Example 3, the candidate set is the singleton $\Delta(\mathcal{O}) = \{\{vro, ssc\}\}$, where $\{vro, ssc\} = \delta(T)$.

4 WATCHER

The notion of observability of a DES introduced in Definition 5 requires the diagnosis task to match trajectories of $\mathcal{X}$ with regular languages specified by regular expressions. Based on Equation (5), a candidate in $\Delta(\mathcal{O})$ is the diagnosis of a trajectory T such that $\mathcal{O} \in T^*$. Based on Definition 6, checking whether $\mathcal{O} \in T^*$ means that we need to understand when observations occur based on the sequence of component transitions in T. Specifically, for each $(\mathbb{T}, \mathcal{L}, o) \in Obs(\mathcal{X})$, at any point of a prefix T_i of T, namely $T_i = [t_1, \ldots, t_i]$, we need to check if the restriction on $\mathbb{T}$ of a suffix of T_i is a string in $\mathcal{L}$. If so, the observation o should be in a proper position in $\mathcal{O}$ (otherwise, T does not conform with $\mathcal{O}$). The critical point is therefore to keep tracking possible strings in $\mathcal{L}$ based on sequences of component transitions in T. Since $\mathcal{L}$ is regular, it can be recognized by a finite automaton. However, a simple recognizer is not sufficient for the task, as strings of the same language may overlap in T. To cope with possibly overlapping strings in the languages associated with observations, the notion of a *watcher* is introduced.

Definition 8 (Watcher)**.** Let $\mathcal{X}$ be a DES, let $\mathbf{T}$ be the set of component transitions in $\mathcal{X}$, and let $(\mathbb{T}, \mathcal{L}, o) \in Obs(\mathcal{X})$. Let $\mathcal{R}_o$ be the *recognizer* of o, a finite automaton recognizing $\mathcal{L}$. Let $\mathcal{R}_o^\epsilon$ be the NFA obtained from $\mathcal{R}_o$ by inserting an ϵ-transition from each non-initial state to the initial state. The *watcher* of o, namely $\mathcal{W}_o$, is a DFA that is obtained by the determinization of $\mathcal{R}_o^\epsilon$.

Example 7. With reference to $Obs(\mathcal{V})$ defined in Example 4 (cf. Table 3), consider the language $\mathcal{L} = v_3\, v_3^+ \mid v_4\, v_4^+$, which is associated with the observation *stk*. Shown in Figure 3 are the recognizer $\mathcal{R}_{stk}$ (left), the NFA $\mathcal{R}_{stk}^\epsilon$ (center), and the

watcher $\mathcal{W}_{stk}$ (right). The watchers $\mathcal{W}_{sns}, \mathcal{W}_{vlv}, \mathcal{W}_{stk}$, and $\mathcal{W}_{bal}$, corresponding to the observations defined in Table 3, are shown in Figure 4.

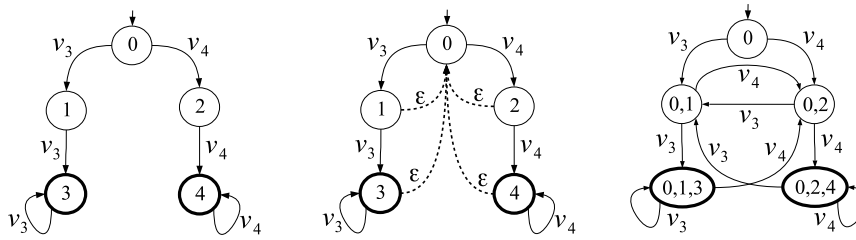

Figure 3. From left to right: $\mathcal{R}_{stk}$, $\mathcal{R}^{\epsilon}_{stk}$, and the watcher $\mathcal{W}_{stk}$ (right)

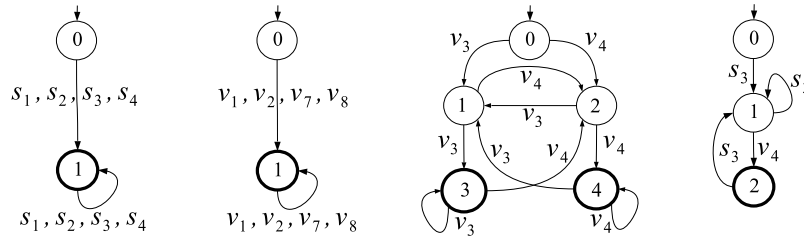

Figure 4. From left to right: watchers $\mathcal{W}_{sns}$, $\mathcal{W}_{vlv}$, $\mathcal{W}_{stk}$, and $\mathcal{W}_{bal}$

The ϵ-transitions in $\mathcal{R}^{\epsilon}$ (cf. Definition 8) allow for a continuous matching of (possibly overlapping) strings, which is in general impossible using a recognizer. To clarify, assume a new abstract observation *abs* defined by the regular language $\mathcal{L}_{abs} = v_3\, v_4 \,|\, v_4\, v_3$, with alphabet $\{v_3, v_4\}$, and the following (hypothetical) trajectory in $\mathcal{V}^*$:

$$T = \left[s_3, \overbrace{v_4, \underbrace{v_3}, v_4}^{T'}, s_2 \right]_{T''} . \tag{6}$$

As such, T includes two overlapping subtrajectories in $\mathcal{L}_{abs}$, namely $T' = [\, v_4, v_3 \,]$ and $T'' = [\, v_3, v_4 \,]$, where the last transition v_3 of T' is the first transition of T''. Hence, the observation *abs* is emitted twice in T, namely at the last transition of T' and T'', respectively. Assume further to trace the emission of *abs* based on the recognizer $\mathcal{R}_{abs}$ (cf. Figure 5, left). When the final state 4 is reached, *abs* is emitted. At this point, since no transition exits the final state 4, the recognizer starts again from the initial state 0 in order to keep matching. It first changes state to 2 in correspondence of v_4, and with s_2 (mismatch) it returns to 0. The result is that, owing to the overlapping of the subtrajectories T' and T'', the second emission of *abs* is undetected. By contrast, consider matching T based on the watcher $\mathcal{W}_{abs}$ (cf.

Figure 5, right). After the detection of *abs* at the final state 4, the next transition v_4 moves to 3, the other final state, thereby also detecting the emission of the second occurrence of *abs*.

Watchers are part of the knowledge that is compiled offline. They are exploited by the diagnosis engine being operated online for matching trajectories with a temporal observation to solve a given diagnosis problem, as clarified in Section 5.

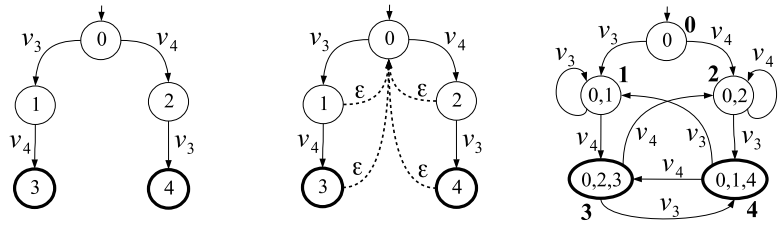

Figure 5. From left to right: $\mathcal{R}_{abs}$, $\mathcal{R}^{\epsilon}_{abs}$, and watcher $\mathcal{W}_{abs}$

5 SOLVING DIAGNOSIS PROBLEMS

The definition of a candidate set $\Delta(\mathcal{O})$ provided in Equation (5) is declarative in nature. In other words, no operational actions are suggested for the solution of a diagnosis problem in Definition 7. Worse still, the assumption of the availability of the behavior space $\mathcal{X}^*$ is in general unrealistic, owing to the exponential explosion of the number of system states. Consequently, determining $\Delta(\mathcal{O})$ becomes a computational issue where the soundness and completeness of the set of candidates is required under the assumption that $\mathcal{X}^*$ is missing. The idea is therefore to generate (online) the subspace of $\mathcal{X}^*$ comprising exactly the trajectories T of $\mathcal{X}$ fulfilling the condition $\mathcal{O} \in T^*$ in Equation (5), called the *$\mathcal{O}$-constrained space* of $\mathcal{X}$ (Definition 9). Eventually, the candidates are determined based on a decoration of the constrained space, called the *abduction* of $\mathcal{O}$ (Definition 10).

Definition 9 (Constrained Space)**.** Let $\mathcal{O} = [o_1, \ldots, o_n]$ be a temporal observation of $\mathcal{X}$, let $\mathcal{X}^* = (\Sigma, X, \tau, x_0, X_\mathrm{f})$ be the behavior space of $\mathcal{X}$, let $Obs(\mathcal{X}) = \{(\mathbb{T}_1, \mathcal{L}_1, o'_1), \ldots, (\mathbb{T}_k, \mathcal{L}_k, o'_k)\}$ be the observability of $\mathcal{X}$, and let $\mathcal{W}_i = (\mathbb{T}_i, W_i, \tau_i, w_{0i}, W_{\mathrm{f}i})$ be the watcher of o'_i, $i \in [1\,..\,k]$. The *$\mathcal{O}$-constrained space* of $\mathcal{X}$ is a DFA

$$\mathcal{X}^*_{\mathcal{O}} = (\Sigma, Y, \tau_y, y_0, Y_\mathrm{f}) \tag{7}$$

where $Y \subseteq X \times W \times [0\,..\,n]$ is the set of states, where $W = (W_1 \times \cdots \times W_k)$, $y_0 = (x_0, w_0, 0)$ is the initial state, where $w_0 = (w_{01}, \ldots, w_{0k})$, $Y_\mathrm{f} \subseteq Y$ is the set of final states, with $(x, w, \Im) \in Y_\mathrm{f}$ when $x \in X_\mathrm{f}$ and $\Im = n$, and $\tau_y : Y \times \Sigma \mapsto Y$ is the transition function, with $\tau_y((x, w, \Im), t) = (x', w', \Im')$, $w = (w_1, \ldots, w_k)$, $w' = (w'_1, \ldots, w'_k)$, when $\tau(x, t) = x'$, $(x', w', \Im')$ is connected with a state in Y_f,

$\forall i \in [1\,..\,k]$:

$$w_i' = \begin{cases} \bar{w}_i, & \text{if } t \in \mathbb{T}_i \text{ and } \tau_i(w_i, t) = \bar{w}_i \\ w_{0i}, & \text{if } t \in \mathbb{T}_i \text{ and } \tau_i(w_i, t) \text{ is undefined,} \\ w_i, & \text{if } t \notin \mathbb{T}_i, \end{cases} \tag{8}$$

and, let $O = \{o_i' \mid i \in [1\,..\,k], \tau_i(w_i, t) = w_i', w_i' \in W_{\mathrm{f}i}\}$, $|O|$ denoting the cardinality of O, we have $\Im' = \Im + |O|$, provided that $\Im' \leq n$ and $O = \{\, o_{\Im+j} \mid j \in [1\,..\,|O|]\,\}$.

One may argue that Definition 9 assumes the availability of the behavior space $\mathcal{X}^*$, thereby contradicting the assumption of its unavailability. In reality, the reference to $\mathcal{X}^*$ is handy for formal reasons only. The construction of $\mathcal{X}^*_{\mathcal{O}}$ can (and will) be performed without $\mathcal{X}^*$, by reasoning on the model of $\mathcal{X}$ and applying the triggerable component transitions for generating the system states, where a state is final when all links are empty. For instance, the condition $\tau(x, t) = x'$ is translated into checking the triggerability of the component transition t at the system state x and, if so, by generating the new state $x' = (S', E')$ as an updated copy of x by setting in S' the new component state relevant to t, by removing from E' the input event of t, and by inserting into E' the output events of t (if any). In this generation, it is essential to capture the occurrences of observations (cf. Equation (8)) and to match them against $\mathcal{O}$ by comparing the set O of observations occurring at t with the next observations in $\mathcal{O}$, irrespective of their temporal ordering (cf. Definition 6). Eventually, only the states that are involved in a trajectory of $\mathcal{X}$ are retained (namely, those connected with a final state).

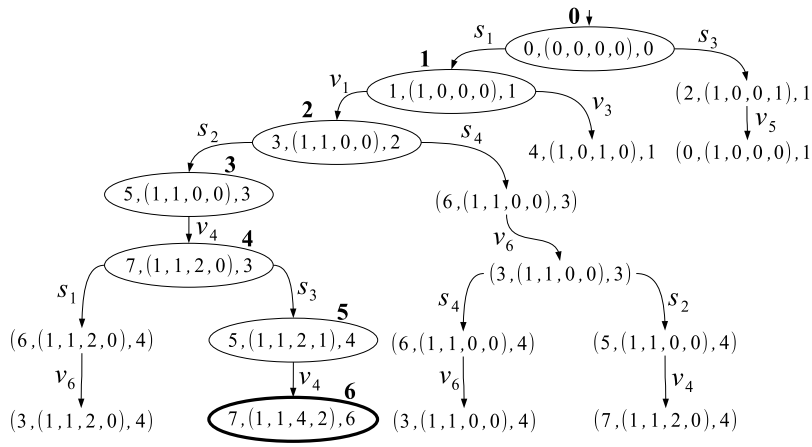

Figure 6. Generation of the $\mathcal{O}$-constrained space $\mathcal{V}^*_{\mathcal{O}}$ of the thermovalve, where $\mathcal{O} = [sns, vlv, sns, sns, stk, bal]$

Example 8. Shown in Figure 6 is the generation of the $\mathcal{O}$-constrained space $\mathcal{V}^*_{\mathcal{O}}$ of the thermovalve, where $\mathcal{O} = [sns, vlv, sns, sns, stk, bal]$ (cf. Example 6). The actual states of $\mathcal{V}^*_{\mathcal{O}}$ are circled and renamed $0\,..\,6$, where 6 is the only final state. The other states (those not circled) are discarded because they are not connected with any final state, thus being not part of any trajectory of $\mathcal{V}$. Based on Definition 9, each state in $\mathcal{X}^*_{\mathcal{O}}$ is identified by a triple $(\nu, w, \Im)$, where ν is a state of $\mathcal{V}$ (cf. Figure 2), $w = (w_{sns}, w_{vlv}, w_{stk}, w_{bal})$ is a quadruple of states of the watchers W_{sns}, W_{vlv}, W_{stk}, and W_{bal} (cf. Figure 4), and $\Im \in [0\,..\,6]$ is the *index* of $\mathcal{O}$, indicating the prefix of $\mathcal{O}$ that has been matched already. In summary, $\mathcal{V}^*_{\mathcal{O}}$ includes one trajectory only, namely $T = [s_1, v_1, s_2, v_4, s_3, v_4]$.

For efficiency reasons, the field w within a state of $\mathcal{X}^*_{\mathcal{O}}$ may comprise only the states of the watchers relevant to the languages in $Obs(\mathcal{X})$ including sequences of two or more transitions (in Example 8, $\mathcal{W}_{stk}$ and $\mathcal{W}_{bal}$). So, the states of the watchers relevant to the languages including strings of single transitions only may be disregarded. In Example 8, the missing watchers would be $\mathcal{W}_{sns}$ and $\mathcal{W}_{vlv}$, which are relevant to the regular languages $\mathcal{L}_{sns} = \{[s_1], [s_2], [s_3], [s_4]\}$ and $\mathcal{L}_{vlv} = \{[v_1], [v_2], [v_7], [v_8]\}$, respectively (cf. Table 3). The rationale behind this simplification lies in that the observation associated with single component transitions can be detected directly based on the component transition t marking each transition $\langle (x, w, \Im), t, (x', w', \Im) \rangle$ in $\mathcal{X}^*_{\mathcal{O}}$. For instance, the transition $\langle 0, s_1, 1 \rangle$ in $\mathcal{V}^*_{\mathcal{O}}$ (cf. Figure 6) allows for detecting the observation *sns* regardless of the state of $\mathcal{W}_{sns}$.

Proposition 1. The language of an $\mathcal{O}$-constrained space $\mathcal{X}^*_{\mathcal{O}}$ is the sublanguage of $\mathcal{X}^*$ comprising the trajectories that conform with the temporal observation $\mathcal{O}$, namely

$$\{\, T \mid T \in \mathcal{X}^*_{\mathcal{O}} \,\} = \{\, T \mid T \in \mathcal{X}^*, \mathcal{O} \in T^* \,\}\,. \tag{9}$$

Proof.

Soundness. If $T \in \mathcal{X}^*_{\mathcal{O}}$, then $T \in \mathcal{X}^*$, $\mathcal{O} \in T^*$. Based on Definition 9, $\tau_y((x, w, \Im), t) = (x', w', \Im')$ in $\mathcal{X}^*_{\mathcal{O}}$ requires $\tau(x, t) = x'$ in $\mathcal{X}^*$; hence, $T \in \mathcal{X}^*$. We have to show that $\mathcal{O} \in T^*$ also. Based on Definition 6, given $T = [t_1, \ldots, t_n]$, a sequence $\mathbb{O} = [O_1, \ldots, O_n]$ is defined, where each $O_i \in \mathbb{O}$, $i \in [1\,..\,n]$, is a (possibly empty) set of observations o such that, given a suffix T' of the prefix $[t_1, \ldots, t_i]$ of T, we have $T'_{[\mathbb{T}]} \in \mathcal{L}$, where $(\mathbb{T}, \mathcal{L}, o) \in Obs(\mathcal{X})$. Any sequence of observations obtained by transforming each set in $\mathbb{O}$ into a sequence and by concatenating all these sequences is a temporal observation $\mathcal{O} \in T^*$. Based on Definition 9, the new states of the watchers in w', defined in Equation (8), allow for the computation of a set O of observations that equals a corresponding set in $\mathbb{O}$. In fact, O is matched against the next observations in $\mathcal{O}$, namely $o_{\Im+1}, \ldots, o_{\Im+|O|}$, where $|O|$ is the cardinality of O. Eventually, this matching guarantees that $\mathcal{O}$ can be generated by concatenating all $\bar{O} \in \bar{\mathbb{O}}$, where each $\bar{O}$ is obtained by transforming a set $O \in \mathbb{O}$ into a sequence.

Completeness. If $T \in \mathcal{X}^*$, $\mathcal{O} \in T^*$, then $T \in \mathcal{X}^*_{\mathcal{O}}$. Based on Definition 9, $\tau(x,t) = x'$ in $\mathcal{X}^*$ is a requisite for the definition of $\tau_y((x,w,\Im),t) = (x',w',\Im')$ in $\mathcal{X}^*_{\mathcal{O}}$. We have to show that the additional constraint $\mathcal{O} \in T^*$ is valid for $\mathcal{X}^*_{\mathcal{O}}$ also. Based on Definition 6, $\mathcal{O} \in T^*$ means that $\mathcal{O}$ can be generated by concatenating the sequences in $\bar{\mathbb{O}}$ obtained from each set of observations $O_i \in \mathbb{O}$, $i \in [1\,..\,n]$, where each O_i includes the observations o such that $(\mathbb{T},\mathcal{L},o) \in Obs(\mathcal{X})$, T' is a suffix of $[t_1,\ldots,t_i]$, and $T'_{[\mathbb{T}]} \in \mathcal{L}$. Since, for the tuple w' of watcher states, the same O_i is generated based on the final states of the watchers, the subsequent conditions on the matching of O_i against the next observations in $\mathcal{O}$, namely $o_{\Im+1},\ldots,o_{\Im+|O_i|}$, with $|O_i|$ being the cardinality of O_i, are fulfilled for all transitions in $t_i \in T$. Hence, $T \in \mathcal{X}^*_{\mathcal{O}}$.

□

According to Proposition 1, any trajectory $T \in \mathcal{X}^*_{\mathcal{O}}$ conforms with $\mathcal{O}$ and no other trajectory does. Hence, based on Definition 7, the set of candidates $\Delta(\mathcal{O})$ can be determined by providing the set of diagnoses $\delta(T)$ such that $T \in \mathcal{X}^*_{\mathcal{O}}$.

Definition 10 (Abduction)**.** Let $\mathcal{O}$ be a temporal observation of a DES $\mathcal{X}$. The *abduction* of $\mathcal{O}$ is a DFA $\mathcal{X}^{\Delta}_{\mathcal{O}}$ obtained from the $\mathcal{O}$-constrained space

$$\mathcal{X}^*_{\mathcal{O}} = (\Sigma, Y, \tau_y, y_0, Y_{\mathrm{f}}) \tag{10}$$

by marking each state $y \in Y$ with a set of diagnoses $\Delta(y)$ as follows:

1. For the initial state y_0, $\emptyset \in \Delta(y_0)$;
2. For each transition $\langle y,t,y'\rangle$, for each $\delta \in \Delta(y)$, if $(t,f) \in Abn(\mathcal{X})$, then $\delta \cup \{f\} \in \Delta(y')$, else $\delta \in \Delta(y')$.

Based on the first rule in Definition 10, the empty diagnosis corresponds to the empty semi-trajectory. Based on the second (inductive) rule, if the decoration of a state y includes a diagnosis δ, then there is at least one semi-trajectory T, ending at y, whose diagnosis is δ. Thus, there is a semi-trajectory $T \cup [\,t\,]$ ending at y' whose diagnosis is either the extension of δ by the fault f associated with the component transition t in $Abn(\mathcal{X})$, when $(t,f) \in Abn(\mathcal{X})$, or δ, when $(t,f) \notin Abn(\mathcal{X})$. Note that, even when the number of trajectories in $\mathcal{X}^*_{\mathcal{O}}$ is infinite (owing to cycles), since a diagnosis is a set, the faults involved in a cycle are not duplicated. Hence, the abduction can be generated in a finite number of decoration actions. These considerations support Proposition 2.

Proposition 2. Let T be a trajectory of an abduction $\mathcal{X}^{\Delta}_{\mathcal{O}}$ ending at a final state y. We have $\delta(T) \in \Delta(y)$. Conversely, if $\bar{\delta} \in \Delta(y)$, where y is a final state of $\mathcal{X}^{\Delta}_{\mathcal{O}}$, then there is a trajectory $T \in \mathcal{X}^{\Delta}_{\mathcal{O}}$ such that $\bar{\delta} = \delta(T)$.

Example 9. Shown in Figure 7 is the abduction $\mathcal{V}_{\mathcal{O}}^{\Delta}$ relevant to the $\mathcal{O}$-constrained space $\mathcal{V}_{\mathcal{O}}^{*}$ shown in Figure 6 (cf. the abnormality $Abn(\mathcal{V})$ listed in Table 2). Notably, the set of diagnoses marking the final state, namely $\{\{vro, ssc\}\}$, equals the candidate set $\Delta(\mathcal{O})$ determined in Example 6 based on the same temporal observation $\mathcal{O}$ of $\mathcal{V}$. This is no coincidence, as formally claimed in Theorem 1.

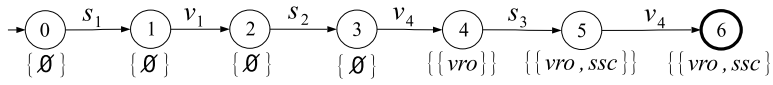

Figure 7. Abduction $\mathcal{V}_{\mathcal{O}}^{\Delta}$, where $\mathcal{O} = [sns, vlv, sns, sns, stk, bal]$

Theorem 1. The candidate set of a temporal observation $\mathcal{O}$ of $\mathcal{X}$ can be determined based on the set Y_{f} of final states of the abduction $\mathcal{X}_{\mathcal{O}}^{\Delta}$, specifically

$$\Delta(\mathcal{O}) = \bigcup_{y \in Y_{\mathrm{f}}} \Delta(y). \tag{11}$$

Proof. Let Δ_y denote $\bigcup_{y \in Y_{\mathrm{f}}} \Delta(y)$ in Equation (11). We have to show that, if $\bar{\delta} \in \Delta(\mathcal{O})$, then $\bar{\delta} \in \Delta_y$ (completeness); and, if $\bar{\delta} \in \Delta_y$, then $\bar{\delta} \in \Delta(\mathcal{O})$ (soundness). If $\bar{\delta} \in \Delta(\mathcal{O})$, then, based on Equation (5), there is a trajectory $T \in \mathcal{X}^*$ such that $\mathcal{O} \in T^*$. Hence, according to Equation (9) of Proposition 1, T is also in $\mathcal{X}_{\mathcal{O}}^{*}$ (as well as in $\mathcal{X}_{\mathcal{O}}^{\Delta}$); besides, based on Proposition 2, Δ_y includes $\bar{\delta}$. On the other hand, if $\bar{\delta} \in \Delta_y$, then, according to Proposition 2, there is a trajectory $T \in \mathcal{X}_{\mathcal{O}}^{\Delta}$ (as well as in $\mathcal{X}_{\mathcal{O}}^{*}$) such that $\bar{\delta} = \delta(T)$. According to Equation (9) of Proposition 1, $T \in \mathcal{X}^*$ and $\mathcal{O} \in T^*$. Hence, based on Equation (5) of Definition 7, $\bar{\delta} \in \Delta(\mathcal{O})$. □

6 OBSERVABILITY AND UNCERTAINTY

The input of a DES diagnosis task is usually a temporal observation, namely a totally temporally ordered sequence of observations, where it is assumed that an observed event equals the emitted event and the reception order of events equals the emission order. Still, as proposed in [20], the temporal observation may become *uncertain* in nature. An *uncertain temporal observation* is represented as a directed acyclic graph, where each vertex is an observed event and each edge is a (partial) temporal precedence relationship. Partial temporal ordering leads to *temporal uncertainty*, which may be caused either by the lack of timestamps associated with observations or by unreliable timestamps. Besides, each vertex comprises a *set* of observations, possibly including the *null* observation (denoting no observation), of which only one is the (unknown) actual observation emitted by the DES. This is called *logical uncertainty*, which may be caused by noise in the communication channel(s) connecting the DES with the observer. Now, what is the relationship between uncertain observations and generalized observability? The answer is that uncertain temporal

observations are orthogonal to generalized observability, since uncertainty in observations is relevant to the *perception* of the observable events produced by a DES being operated, whereas generalized observability is relevant to the *genesis* of these events.

7 CONCLUSION

Observability has received little (if any) attention in model-based diagnosis of DESs. This may sound striking if we consider the amount of attention devoted to other DES properties, such as abnormality and, overwhelmingly more so, diagnosability. Inspired by humans observing reality, where a combination of events may be registered by the mind as a single perception, this paper has proposed a diagnosis technique for DESs based on a generalized notion of observability. What is observed is no longer confined to a single transition; rather, it is extended to a (possibly large) fragment of the DES behavior. This comes with a price, since the detection of the (abstract) observations requires the diagnosis engine to store the states of the watchers within the states of the $\mathcal{O}$-constrained space of the DES. To alleviate this problem, knowledge compilation techniques may be designed for constructing (offline) extended diagnosers, possibly constructed incrementally [6]. Finally, the integration with abstract abnormality [22, 26] and the adaptation with complex ASs [19, 23] as well as deep DESs [25], are exciting topics for future research.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61972360).

REFERENCES

[1] Baroni, P.—Lamperti, G.—Pogliano, P.—Zanella, M.: Diagnosis of Large Active Systems. Artificial Intelligence, Vol. 110, 1999, No. 1, pp. 135–183, doi: 10.1016/S0004-3702(99)00019-3.

[2] Basile, F.: Overview of Fault Diagnosis Methods Based on Petri Net Models. Proceedings of the 2014 European Control Conference (ECC 2014), 2014, pp. 2636–2642, doi: 10.1109/ECC.2014.6862631.

[3] Bertoglio, N.—Lamperti, G.—Zanella, M.—Zhao, X.: Diagnosis of Temporal Faults in Discrete-Event Systems. In: De Giacomo, G., Catala, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (Eds.): 24th European Conference on Artificial Intelligence (ECAI 2020). IOS Press, Amsterdam, Frontiers in Artificial Intelligence and Applications, Vol. 325, 2020, pp. 632–639, doi: 10.3233/FAIA200148.

[4] Bertoglio, N.—Lamperti, G.—Zanella, M.—Zhao, X.: Explanatory Diagnosis of Discrete-Event Systems with Temporal Information and Smart Knowledge-

Compilation. In: Calvanese, D., Erdem, E., Thielsher, M. (Eds.): Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020). IJCAI Organization, 2020, pp. 130–140, doi: 10.24963/kr.2020/14.

[5] Bertoglio, N.—Lamperti, G.—Zanella, M.—Zhao, X.: Explanatory Monitoring of Discrete-Event Systems. In: Czarnowski, I., Howlett, R., Jain, L. (Eds.): Intelligent Decision Technologies (IDT 2020). Springer, Singapore, Smart Innovation, Systems and Technologies, Vol. 193, 2020, pp. 63–77, doi: 10.1007/978-981-15-5925-9_6.

[6] Bertoglio, N.—Lamperti, G.—Zanella, M.—Zhao, X.: Temporal-Fault Diagnosis for Critical-Decision Making in Discrete-Event Systems. In: Cristani, M., Toro, C., Zanni-Merk, C., Howlett, R., Jain, L. (Eds.): Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 24th International Conference KES2020. Procedia Computer Science, Vol. 176, 2020, pp. 521–530, doi: 10.1016/j.procs.2020.08.054.

[7] Brand, D.—Zafiropulo, P.: On Communicating Finite-State Machines. Journal of the ACM, Vol. 30, 1983, No. 2, pp. 323–342, doi: 10.1145/322374.322380.

[8] Cabasino, M. P.—Giua, A.—Seatzu, C.: Fault Detection for Discrete Event Systems Using Petri Nets with Unobservable Transitions. Automatica, Vol. 46, 2010, No. 9, pp. 1531–1539, doi: 10.1016/j.automatica.2010.06.013.

[9] Cassandras, C.—Lafortune, S.: Introduction to Discrete Event Systems. Second Edition. Springer, New York, 2008, doi: 10.1007/978-0-387-68612-7.

[10] Cong, X.—Fanti, M. P.—Mangini, A. M.—Li, Z.: Decentralized Diagnosis by Petri Nets and Integer Linear Programming. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 48, 2018, No. 10, pp. 1689–1700, doi: 10.1109/TSMC.2017.2726108.

[11] Debouk, R.—Lafortune, S.—Teneketzis, D.: Coordinated Decentralized Protocols for Failure Diagnosis of Discrete-Event Systems. Discrete Event Dynamic Systems: Theory and Applications, Vol. 10, 2000, No. 1–2, pp. 33–86, doi: 10.1023/A:1008335115538.

[12] Grastien, A.—Cordier, M.-O.—Largouët, C.: Incremental Diagnosis of Discrete-Event Systems. Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinburgh, UK, 2005, pp. 1564–1565.

[13] Grastien, A.—Haslum, P.—Thiébaux, S.: Conflict-Based Diagnosis of Discrete Event Systems: Theory and Practice. Thirteenth International Conference on Knowledge Representation and Reasoning (KR 2012), Rome, Italy, Association for the Advancement of Artificial Intelligence, 2012, pp. 489–499.

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

[15] Jéron, T.—Marchand, H.—Pinchinat, S.—Cordier, M.-O.: Supervision Patterns in Discrete Event Systems Diagnosis. 2006 8th International Workshop on Discrete Event Systems (WODES 2006), IEEE, 2006, pp. 262–268, doi: 10.1109/WODES.2006.1678440.

[16] Jiroveanu, G.—Boel, R. K.—Bordbar, B.: On-Line Monitoring of Large Petri Net Models Under Partial Observation. Discrete Event Dynamic Systems: Theory and Applications, Vol. 18, 2008, No. 3, pp. 323–354, doi: 10.1007/s10626-007-0036-x.

[17] Kan John, P.—Grastien, A.: Local Consistency and Junction Tree for Diagnosis of Discrete-Event Systems. In: Ghallab, M., Spyropoulos, C. D., Fakotakis, N., Avouris, N. (Eds.): Eighteenth European Conference on Artificial Intelligence (ECAI 2008). IOS Press, Amsterdam, Frontiers in Artificial Intelligence and Applications, Vol. 178, 2008, pp. 209–213, doi: 10.3233/978-1-58603-891-5-209.

[18] Kwong, R. H.—Yonge-Mallo, D. L.: Fault Diagnosis in Discrete-Event Systems: Incomplete Models and Learning. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 41, 2011, No. 1, pp. 118–130, doi: 10.1109/TSMCB.2010.2047257.

[19] Lamperti, G.—Quarenghi, G.: Intelligent Monitoring of Complex Discrete-Event Systems. In: Czarnowski, I., Caballero, A., Howlett, R., Jain, L. (Eds.): Intelligent Decision Technologies 2016 (IDT 2016). Springer, Cham, Smart Innovation, Systems and Technologies, Vol. 56, 2016, pp. 215–229, doi: 10.1007/978-3-319-39630-9_18.

[20] Lamperti, G.—Zanella, M.: Diagnosis of Discrete-Event Systems from Uncertain Temporal Observations. Artificial Intelligence, Vol. 137, 2002, No. 1–2, pp. 91–163, doi: 10.1016/S0004-3702(02)00123-6.

[21] Lamperti, G.—Zanella, M.: Flexible Diagnosis of Discrete-Event Systems by Similarity-Based Reasoning Techniques. Artificial Intelligence, Vol. 170, 2006, No. 3, pp. 232–297, doi: 10.1016/j.artint.2005.08.002.

[22] Lamperti, G.—Zanella, M.: Context-Sensitive Diagnosis of Discrete-Event Systems. In: Walsh, T. (Ed.): Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), Vol. 2, AAAI Press, 2011, pp. 969–975.

[23] Lamperti, G.—Zanella, M.—Zhao, X.: 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 (KR 2018). AAAI Press, 2018, pp. 464–473.

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

[25] Lamperti, G.—Zanella, M.—Zhao, X.: Diagnosis of Deep Discrete-Event Systems. Journal of Artificial Intelligence Research, Vol. 69, 2020, pp. 1473–1532, doi: 10.1613/jair.1.12171.

[26] Lamperti, G.—Zhao, X.: Diagnosis of Active Systems by Semantic Patterns. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 44, 2014, No. 8, pp. 1028–1043, doi: 10.1109/TSMC.2013.2296277.

[27] Pencolé, Y.—Cordier, M.-O.: A Formal Framework for the Decentralized Diagnosis of Large Scale Discrete Event Systems and Its Application to Telecommunication Networks. Artificial Intelligence, Vol. 164, 2005, No. 1–2, pp. 121–170, doi: 10.1016/j.artint.2005.01.002.

[28] Pencolé, Y.—Steinbauer, G.—Mühlbacher, C.—Travé-Massuyès, L.: Diagnosing Discrete Event Systems Using Nominal Models Only. In: Zanella, M.,

Pill, I., Cimatti, A. (Eds.): 28th International Workshop on Principles of Diagnosis (DX'17). EasyChair, Kalpa Publications in Computing, Vol. 4, 2018, pp. 169–183, doi: 10.29007/1d2x.

[29] Reiter, R.: A Theory of Diagnosis from First Principles. Artificial Intelligence, Vol. 32, 1987, No. 1, pp. 57–95, doi: 10.1016/0004-3702(87)90062-2.

[30] Sampath, M.—Sengupta, R.—Lafortune, S.—Sinnamohideen, K.—Teneketzis, D.: Diagnosability of Discrete-Event Systems. IEEE Transactions on Automatic Control, Vol. 40, 1995, No. 9, pp. 1555–1575, doi: 10.1109/9.412626.

[31] Sampath, M.—Sengupta, R.—Lafortune, S.—Sinnamohideen, K.—Teneketzis, D.: Failure Diagnosis Using Discrete-Event Models. IEEE Transactions on Control Systems Technology, Vol. 4, 1996, No. 2, pp. 105–124, doi: 10.1109/87.486338.

[32] Struss, P.: Fundamentals of Model-Based Diagnosis of Dynamic Systems. Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI 1997), Nagoya, Japan, 1997, pp. 480–485.

Gianfranco Lamperti received the Doctoral degree in electronic engineering (computer science) from Politecnico di Milano, Italy, in 1986. For several years, he did research in the private sector in areas related to formal languages, databases, and artificial intelligence. He is currently Associate Professor of computer science with the Department of Information Engineering, University of Brescia, Italy. His main research interests include engineering issues in finite automata and model-based diagnosis of discrete-event systems.

Marina Zanella received her M.Sc. degree in electronics engineering from Politecnico di Milano, Italy, in 1986. She is currently Associate Professor with the Department of Information Engineering, University of Brescia, Italy. Her research interests include model-based reasoning for monitoring and diagnosis of static systems and discrete-event systems, diagnosability analysis, and uncertain-knowledge modeling. She is the co-author of three books and over 130 scientific papers.

Xiangfu Zhao received the Doctoral degree in computer science and technology from Jilin University, Changchun, China, in 2009. He is currently Professor of computer science with the School of Computer and Control Engineering, Yantai University, China. His main research interests include model-based diagnosis, discrete-event systems, and blockchain. He is the co-author of more than 50 research papers.