HAL open science

Discriminability Analysis of Supervision Patterns by Net Unfoldings

Houssam Eddine Gougam, Audine Subias, Yannick Pencolé

▶ To cite this version:

Houssam Eddine Gougam, Audine Subias, Yannick Pencolé. Discriminability Analysis of Supervision Patterns by Net Unfoldings. 12th IFAC International Workshop on Discrete Event Systems (2014)-IFAC Proceedings Volumes, May 2014, Cachan, France. pp.459 - 464, 10.3182/20140514-3-FR-4046.00136 . hal-01847155

HAL Id: hal-01847155
https://laas.hal.science/hal-01847155

Submitted on 23 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discriminability Analysis of Supervision Patterns by Net Unfoldings

Houssam-Eddine Gougam [*] **Audine Subias** [*] **Yannick Pencolé** [†]

[*] *CNRS; LAAS; 7 avenue du Colonel Roche, F-31400 Toulouse, France*
Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
[†] *CNRS; LAAS; 7 avenue du Colonel Roche, F-31400 Toulouse, France*
Univ de Toulouse, LAAS, F-31400 Toulouse, France

Abstract: In this paper, we are interested in the discriminability of supervision patterns, in discrete event systems (DES). Discriminability — as opposed to diagnosability — is the possibility to detect the *exclusive* occurrence of a particular behavior of interest — called the supervision pattern. To this end, we propose to adapt the classical twin-plant approach to Petri nets unfolding. The usage of unfoldings permits us to avoid the combinatorial explosion associated with marking graphs. The method can also be used to solve the classical problem of discrete event systems' diagnosability.

Keywords: discriminability, diagnosability, supervision patterns, labeled Petri nets, Petri nets unfolding.

1. INTRODUCTION

In the context of discrete-event systems (DES), the supervision task consists in analyzing the sequence of observed events and determining whether an abnormal/faulty situation has occurred before deciding what kind of actions to perform in order to recover the optimal performance of the system. By the intrinsic nature of DES, an abnormal situation is characterized by a partial order of observable/non-observable events called an *event pattern* or a *supervision pattern* as introduced in Jéron et al. (2006).

In this paper, we address the problem of analysing the discriminability of a set of supervision patterns. Two supervision patterns $\mathcal{R}_1$ and $\mathcal{R}_2$ are discriminable if it is always possible to assert from the observed events that if $\mathcal{R}_1$ (resp. $\mathcal{R}_2$) has occurred then $\mathcal{R}_2$ (resp. $\mathcal{R}_1$) has not occurred and will not occur.

The notion of pattern discriminability is obviously related to the notion of pattern diagnosability and their respective analyses (Jiang et al. (2003), Jéron et al. (2006), Yoo and Garcia (2008), Gougam et al. (2013)), however the point of view is different. The result of a diagnosability analysis states whether any considered pattern is always detectable or not. If such a property holds then it is possible to design a diagnoser that will always be able to determinate which patterns have occurred in a finite amount of time. However there is a pitfall: underlying real-world systems are not and cannot be diagnosable as the diagnosability property is very restrictive (Pencolé (2005)) and requires an observability level that cannot be implemented as a set of sensors on the underlying system (physical and/or cost constraints). The classical diagnosability analyses will then conclude such a real-world system is not diagnosable, so what is next?

To cope with this problem, we propose to develop a more constructive analysis by checking the discriminability of the occurrence of a set of patterns with respect to another disjoint set of patterns. The analysis is constructive in the sense that any partial discriminability result has an outcome that can impact the design of a diagnoser for the analysed system even if the underlying system is not globally diagnosable. And finally, once the proposed discriminability analysis is completed, checking the classical diagnosability analysis is straightforward and for free.

In our proposal, the development of the discriminability analysis relies on two original choices. The first one is the use of Petri net — which were shown to be more appropriate to solve diagnosis problems Lai et al. (2008) — to model patterns as well as the system (Basile et al. (2009), Dotoli et al. (2009)) so that we benefit of the natural way to represent event concurrency. The second one is the use of net unfoldings (McMillan (1995), Esparza et al. (2002), Benveniste et al. (2003)) in order to benefit of a representation as a partial order of events. Unfoldings avoid the explicit enumeration of event sequences that is performed by techniques based on marking graphs like in Cabasino et al. (2012). Our proposed approach allows us to have some interesting properties. First, genericness, meaning that we do not impose particular patterns, we rather define a generic framework for supervision pattern, and every pattern falling in this framework can be used for supervision. Second, the supervision patterns are compact, i.e. only relevant events are included leading to more concise patterns. Finally, reusability, which is a direct consequence of the compactness, the exclusive use of relevant events yields supervision patterns which are independent from the system.

The paper is organized as follows. Section 2 introduces the problem, Section 3 presents Petri nets and their use to model the system and the patterns. The proposed analysis method is detailed in Sections 4 and 5. An example of our method is given in Section 6. Finally, we conclude in Section 7 and give some perspectives.

2. DISCRIMINABILITY OF SUPERVISION PATTERNS

Analyzing the discriminability of a system in the DES context means characterizing the performances of the diagnosis algorithm. A diagnosis algorithm, given an observable sequence produced by the system, is responsible of returning its state, i.e. is the system in a *normal* state or a *faulty* state, and in the latter case, which — faulty — behavior yielded this state. If such algorithm exists, and can determine with *certainty* after a *bounded* number of observations that *only* a particular combination of behaviors occurred, this combination is said to be *discriminable*. On the other hand, if the algorithm can detect the occurrence of a behavior with no information about the possible occurrence of other faulty behaviors, the former is said to be *detectable*.

In this paper, we consider a system modeled by a *language*, faulty behaviors are recognized by *supervision patterns* which are also languages with specific properties. More formally, the problem of supervision pattern discriminability in discrete event systems is then defined in the following context.

Let $\Sigma = \{a, b, \ldots\}$ be a finite set called an alphabet. The Kleene closure of Σ denoted Σ^* is the set of finite sequences — or words — over Σ — including the empty sequence, denoted in the following by λ. $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$ is the set of non-empty finite sequences. A subset $\mathcal{S} \subseteq \Sigma^*$ is called a *language* over Σ. The continuation z of a sequence w in $\mathcal{S}$ is a sequence such that $wz \in \mathcal{S}$. So, the set of w's continuations in $\mathcal{S}$ is defined as $\mathcal{S}/w = \{z \in \Sigma^* : wz \in \mathcal{S}\}$. The classical projection of a sequence $w \in \Sigma^*$ on a subset Σ_p of Σ is denoted $\mathcal{P}_{\Sigma_p}(w)$. Finally, $\|w\|$ denotes the length of the sequence w. The behavior of the system is modeled by the prefix-closed language $\mathcal{S}$ over an alphabet Σ representing the set of events generated by the system. Some of these events are observable, while others are not. The observable events are represented by the set Σ_o and the set of non-observable events is called Σ_u. So, $\Sigma = \Sigma_o \cup \Sigma_u$. We make the assumption that the system is Σ_o-alive, meaning: $\forall w \in \mathcal{S}, \exists z \in \mathcal{S}/w : \mathcal{P}_{\Sigma_o}(z) \neq \lambda$; this hypothesis ensures that the system produces observations with some regularity. A supervision pattern $\mathcal{R}$ is a language over $\Sigma_{\mathcal{R}} \subseteq \Sigma$ that recognizes $\mathcal{R}$-faulty sequences.

Definition 1. Let $\mathfrak{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$ be a set of supervision patterns. A sequence $w \in \Sigma^*$ is said to be $\mathcal{R}_i$-faulty if: $\mathcal{P}_{\Sigma_{\mathcal{R}_i}}(w) \in \mathcal{R}_i$. A sequence $w \in \Sigma^*$ is said to be $\mathfrak{R}$-faulty if: $\forall \mathcal{R} \in \mathfrak{R} : w$ is $\mathcal{R}$-faulty. A sequence $w \in \Sigma^*$ is said to be faulty if: $\exists \mathcal{R} \in \mathfrak{R} : w$ is $\mathcal{R}$-faulty.

Definition 2. Let $\mathfrak{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$ be a set of supervision patterns. Let $\mathfrak{R}^s \in 2^{\mathfrak{R}}$ be a subset of $\mathfrak{R}$. A sequence $w \in \Sigma^*$ is said to be *exclusively* $\mathfrak{R}^s$-faulty in $\mathfrak{R}$ if: $\begin{cases} \forall \mathcal{R} \in \mathfrak{R}^s : w \text{ is } \mathcal{R}\text{-faulty} \\ \forall \mathcal{R} \notin \mathfrak{R}^s : w \text{ is not } \mathcal{R}\text{-faulty} \end{cases}$

We will simply use "exclusively $\mathfrak{R}^s$-faulty" rather than "exclusively $\mathfrak{R}^s$-faulty in $\mathfrak{R}$" when there is no ambiguity.

We can now define the discriminability of a set of supervision patterns.

Definition 3. Given a system $\mathcal{S}$ and a set of supervision patterns $\mathfrak{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$. A subset $\mathfrak{R}^s$ of $\mathfrak{R}$ is said to be discriminable if:

$\exists n \in \mathbb{N}, \forall w$ an exclusively $\mathfrak{R}^s$-faulty word of $\mathcal{S}, \forall z \in \mathcal{S}/w :$
$$\|\mathcal{P}_{\Sigma_o}(z)\| \geq n \implies D$$
where the discriminability condition D is:
$$\forall w^o \in \mathcal{P}_{\Sigma_o}^{-1}(\mathcal{P}_{\Sigma_o}(wz)) : w^o \text{ is exclusively } \mathfrak{R}^s\text{-faulty}$$

This property insures that one can detect with *certainty* that *only* a particular combination of patterns — namely $\mathfrak{R}^s$ — occurred.

Similarly, we define the $\mathfrak{R}$-diagnosability of a system.

Definition 4. Let $\mathfrak{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$ be a set of supervision patterns. A language $\mathcal{S}$ is $\mathfrak{R}$-diagnosable if:

$(\forall \mathcal{R} \in \mathfrak{R})(\exists n \in \mathbb{N}), \forall w$ an $\mathcal{R}$-faulty word of $\mathcal{S}, \forall z \in \mathcal{S}/w :$
$$\|\mathcal{P}_{\Sigma_o}(z)\| \geq n \implies D$$
where the diagnosability condition D is:
$$\forall w^o \in \mathcal{P}_{\Sigma_o}^{-1}(\mathcal{P}_{\Sigma_o}(wz)) : w^o \text{ is } \mathcal{R}\text{-faulty}$$

More intuitively, a language $\mathcal{S}$ is $\mathfrak{R}$-diagnosable if it does not contain two arbitrary long sequences, the first $\mathcal{R}$-faulty, the second non $\mathcal{R}$-faulty, which have the same observable behavior.

3. ON MODELING ASPECTS

To tackle the problem of discriminability, we propose to use labeled Petri nets to model the system and the patterns.

3.1 Labeled Petri nets

Definition 5. A *labeled Petri net* is a tuple $\langle P, T, A, \ell, L, \Sigma\rangle$ where:

- P: a set of places;
- T: a set of transitions with $P \cap T = \varnothing$;
- $A \subseteq (P \times T) \cup (T \times P)$: a binary relation representing arcs between nodes;
- $\ell : P \cup T \to L \cup \Sigma \cup \{\lambda\}$: a labeling function where L is the set of place labels, Σ is the set of transition labels and λ denotes the empty sequence.
  ℓ is naturally extended to markings and sequences of nodes. Let M be a marking: $\ell(M) = \{\ell(p) : p \in M\}$. For $s = s_1 s_2 \ldots \in (P \cup T)^*$, $\ell(s) = \ell(s_1)\ell(s_2)\ldots$.

A *marking* M is a map from P to $\mathbb{N}$ which maps any place p to the number of tokens $M(p)$ contained in it. For the sake of simplicity, a marking may sometimes be denoted as a multiset. For instance, let $P = \{p_1, p_2, p_3\}$, the marking M such that $M(p_1) = 2, M(p_2) = 0$ and $M(p_3) = 1$ can be represented as $M = \{p_1, p_1, p_3\}$. A marked and labeled Petri net is a tuple $\Theta = \langle P, T, A, \ell, L, \Sigma, M_0\rangle$ where $\langle P, T, A, \ell, L, \Sigma\rangle$ is a labeled Petri net and M_0 an initial marking. The current state of a Petri net is defined by its current marking. The set $\bullet t = \{p \in P : (p, t) \in A\}$ is the preset of t and $t\bullet = \{p \in P : (t, p) \in A\}$ is its postset (the preset $\bullet p$ and postset $p\bullet$ of a place p are similarly defined). The transition t is *firable* from a given marking M iff: $\forall p \in \bullet t : M(p) > 0$. Firing t leads to a new marking M' such that $M' = (M \setminus \bullet t) \cup t\bullet$ and which is denoted by $M \xrightarrow{t} M'$. A marking M is reachable if there exists a firing sequence $s = t_0 t_1 \ldots t_n$ such that $M_0 \xrightarrow{t_0} M_1 \xrightarrow{t_1} \cdots \xrightarrow{t_n} M$, we can also write $M_0 \xrightarrow{s} M$. Given a marked and labeled Petri net Θ and a set of final markings Q, the G-type language (see Peterson (1977)) generated by Θ is given by: $\mathcal{L}(\Theta) = \{\ell(s) : s \in T^* \wedge M_0 \xrightarrow{s} M \wedge \exists M' \in Q, M' \subseteq M\}$.

The system is modeled by a Petri net Θ. As the language of the system is prefix-closed, it can be represented as the G-type language of a bounded Petri net ($\exists n \in \mathbb{N}, \forall p \in P : M(p) \leq n$) associated with the set of final markings $Q = \{\varnothing\}$ (any reachable marking being a superset of $\varnothing$).

3.2 Supervision Patterns

The classical framework of DES diagnosis (Sampath et al. (1995)) relies on detecting the simple occurrence of particular — faulty — events. The supervision patterns framework, as proposed in Jiang et al. (2003) and in Jéron et al. (2006), extend this analysis to more complex models, enabling us to express more interesting behaviors, possibly involving several events. We propose to represent a supervision pattern as a labeled Petri net.

Definition 6. A *supervision pattern* is a marked labeled Petri net $\langle P, T, A, \ell, L, \Sigma, M_0\rangle$ such that:

(1) (labeling) $L = \{N, F\}$;
(2) (initialization) $\forall p \in P : M_0(p) > 0 \Rightarrow \ell(p) = N$;
(3) (exclusivity) $\forall p_1, p_2 \in P : \ell(p_1) = N \wedge \ell(p_2) = F \Rightarrow M(p_1) \times M(p_2) = 0$;
(4) (stability) for each reachable marking M, $\exists p \in P, M(p) > 0$ and $\ell(p) = F$ implies that for each successor marking $M' : \exists p' \in P, M'(p') > 0$ and $\ell(p') = F$.
(5) (completeness) for each reachable marking M, $\forall e \in \Sigma$, $\exists t \in T : (\ell(t) = e \wedge \forall p \in \bullet t, M(p) > 0)$;

The first condition ensures that every place is either labeled by N or F. Condition 2 states that in the initial state, each marked place is labeled by N. Condition 3 states that a place labeled N and another one labeled F can not be marked at the same time in a reachable marking, meaning that the pattern can not be recognized and not recognized at the same time. Condition 4 ensures that once the pattern is recognized — the place labeled F is marked — it can not return to a non faulty marking. Condition 5 is the completeness condition: from any reachable marking, the supervision pattern can fire a transition labeled by any event in its alphabet, this condition ensures compact independent pattern (the pattern does not include other events than those relevant to faulty behavior).

Any supervision patterns is associated to the set of final markings Q that contain only one marking $Q = \{\{p \in P : \ell(p) = F\}\}$.

Example: event concurrency Let $E = \{e_1, e_2, \ldots\}$ be a set of events. The pattern describing the concurrent occurrence of these events is the labeled Petri net depicted in Figure 1, with the final marking $Q = \{\{p_0\}\}$. Transition t_0, as it is labeled with λ, is firable as soon as every event of E has occurred.

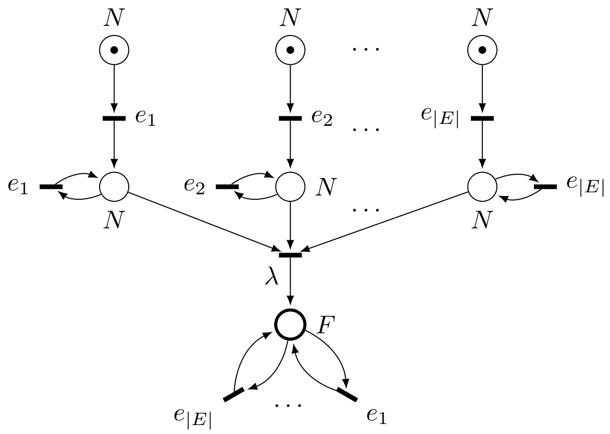


Figure 1. A set of concurrent events.

4. DISCRIMINABILITY ANALYSIS APPROACH

The problem of the discriminability of supervision patterns can now be redefined as being the analysis of the discriminability of a subset of supervision patterns modeled by labeled Petri nets $\mathbf{\Omega}^s \in 2^{\mathbf{\Omega}}$ where $\mathbf{\Omega} = \{\Omega_1, \Omega_2, \ldots, \Omega_n\}$ in a labeled Petri net Θ where $\Sigma_{\Omega_i} \subseteq \Sigma$ for all i.

Definition 7. A subset $\mathbf{\Omega}^s \in 2^{\mathbf{\Omega}}$ is discriminable in $\mathbf{\Omega}$ if $\mathcal{L}(\mathbf{\Omega}^s)$ is discriminable in $\mathcal{L}(\mathbf{\Omega})$. Where:

$$\mathcal{L}(\mathbf{\Omega}) = \{\mathcal{L}(\Omega_1), \mathcal{L}(\Omega_2), \ldots, \mathcal{L}(\Omega_n)\}$$

One classical method used in analyzing the diagnosability of faults in a discrete event system — introduced in Yoo and Lafortune (2002) and Jiang et al. (2000), is the use of a *twin-plant* on the observable events to detect normal and faulty executions that share the same observable behavior — if any — and thus conclude about the non-diagnosability of the system by looking for such sequences that have infinite number of observable events. In the supervision pattern context, the same principle applies. But, first, the supervision patterns must be combined to form a *meta-pattern Π* recognizing all possible combinations of the patterns. Next, the recognition of this meta-pattern must be taken into account within the system Θ. These two operations are realized by two products.

4.1 Synchronized product of labeled Petri nets

Let $\Theta_1 = \langle P_1, T_1, A_1, \ell_1, L_1, \Sigma_1, M_{10}\rangle$ and $\Theta_2 = \langle P_2, T_2, A_2, \ell_2, L_2, \Sigma_2, M_{20}\rangle$ be two labeled Petri nets. The synchronized product $\Theta = \Theta_1 \parallel \Theta_2$ is the labeled Petri net $\Theta = \langle P, T, A, \ell, L, \Sigma, M_0\rangle$ such that:

- $P = P_1 \cup P_2$;
- $T = \hat{T}_1 \cup \hat{T}_2 \cup T_s$,
  where: $\hat{T}_i = \{t \in T_i : \ell_i(t) \notin \Sigma_1 \cap \Sigma_2\}, i = 1,2$; and $T_s = \bigcup_{l \in \Sigma_1 \cap \Sigma_2} \{t_1 \parallel t_2 : \exists (t_1, t_2) \in (T_1 \times T_2) : \ell_1(t_1) = \ell_2(t_2) = l\}$;
- $A = \hat{A}_1 \cup \hat{A}_2 \cup A_s$,
  $\hat{A}_i = A_i \backslash [(P_i \times (T_i \backslash \hat{T}_i)) \cup ((T_i \backslash \hat{T}_i) \times P_i)], i = 1,2;$ and

$$\begin{aligned}A_s = &\{(p,t) \in (P \times T) : p \in P_1, t = t_1 \parallel t_2, (p,t_1) \in A_1\} \\ &\cup \{(p,t) \in (P \times T) : p \in P_2, t = t_1 \parallel t_2, (p,t_2) \in A_2\} \\ &\cup \{(t,p) \in (T \times P) : p \in P_1, t = t_1 \parallel t_2, (t_1,p) \in A_1\} \\ &\cup \{(t,p) \in (T \times P) : p \in P_2, t = t_1 \parallel t_2, (t_2,p) \in A_2\}\end{aligned}$$

- $\ell(p) = \begin{cases} \ell_1(p) & \text{if } p \in P_1 \\ \ell_2(p) & \text{if } p \in P_2 \end{cases}$;
- $\ell(t) = \begin{cases} \ell_1(t_1) & \text{if } t = t_1 \parallel t_2 \in T_s \\ \ell_1(t) & \text{if } t \in T_1 \\ \ell_2(t) & \text{if } t \in T_2 \end{cases}$;
- $L = L_1 \cup L_2$;
- $\Sigma = \Sigma_1 \cup \Sigma_2$;
- $M_0(p) = \begin{cases} M_{10}(p) & \text{if } p \in P_1 \\ M_{20}(p) & \text{if } p \in P_2 \end{cases}$.

This product can be constructed simply by taking the union of the two nets, removing transitions with labels in $\Sigma_1 \cap \Sigma_2$, and for each $(t_1, t_2) \in T_1 \times T_2$ where $\ell(t_1) = \ell(t_2) \neq \lambda$, add a transition $t_1 \parallel t_2$ to the product inheriting their label and arcs.

The product imposes the simultaneous evolution of the two Petri nets, whenever a shared event occurs. In a general context, this can lead to a deadlock — the case where the first petri net needs to fire a synchronized transition while the second one did not reach the appropriate marking. The completeness condition imposed on the supervision patterns ensures that this situation can not happen in our context — the completeness condition states that in any marking, the pattern can fire a transition with any label. So, applying this product on two supervision patterns can model all the possible combinations of their evolution — only one of them evolves, the two evolve in the same time, etc.

Final Markings: Product of Patterns Let Q_i be the set of final markings of $\Omega_i, i = 1, 2$. We define the set of final markings of the product $\Omega = \Omega_1 \parallel \Omega_2$, $Q = Q_1 \cup Q_2$.

Proposition 8. $\mathcal{L}(\Omega) = \{w \in (\Sigma_1 \cup \Sigma_2)^* : \mathcal{P}_{\Sigma_1}(w) \in \mathcal{L}(\Omega_1) \vee \mathcal{P}_{\Sigma_2}(w) \in \mathcal{L}(\Omega_2)\}$.

Proposition 9. $w \in \mathcal{L}(\Omega) \iff \exists w_1 \in \mathcal{L}(\Omega_1), \mathcal{P}_{\Sigma_1 \cap \Sigma_2}(w) = \mathcal{P}_{\Sigma_1 \cap \Sigma_2}(w_1) \vee \exists w_2 \in \mathcal{L}(\Omega_2) : \mathcal{P}_{\Sigma_1 \cap \Sigma_2}(w) = \mathcal{P}_{\Sigma_1 \cap \Sigma_2}(w_2)$.

Final Markings: Product of the System and a Pattern The recognition of the supervision pattern must be taken into account within the system. This is realized by a slight modification of the product — more precisely, the final markings — which is applied to the pattern and the system. For clarity we will use a different symbol for this product $\star$.

Let Q_i be the set of final markings of $\Theta_i, i = 1, 2$. We define the set of final markings of the product $\Theta = \Theta_1 \star \Theta_2$, $Q = \{q = q_1 \cup q_2 : (q_1, q_2) \in Q_1 \times Q_2\}$.

Proposition 10. $\mathcal{L}(\Theta) = \{w \in (\Sigma_1 \cup \Sigma_2)^* : \mathcal{P}_{\Sigma_1}(w) \in \mathcal{L}(\Theta_1) \wedge \mathcal{P}_{\Sigma_2}(w) \in \mathcal{L}(\Theta_2)\}$.

Proposition 11. $w \in \mathcal{L}(\Theta) \iff \exists w_1 \in \mathcal{L}(\Theta_1), \exists w_2 \in \mathcal{L}(\Theta_2) : \mathcal{P}_{\Sigma_1 \cap \Sigma_1}(w) = \mathcal{P}_{\Sigma_1 \cap \Sigma_2}(w) = \mathcal{P}_{\Sigma_1 \cap \Sigma_2}(w_2)$.

From these propositions, we can characterize faulty sequences.

Proposition 12. Given a system Θ and a meta-pattern Π:
$$w \in \mathcal{L}(\Theta) \text{ is faulty} \iff w \in \mathcal{L}(\Theta \star \Pi).$$

All necessary notations being introduced, the analysis method is detailed in the following.

4.2 Analysis stages

We define an "′" operator applicable to sequences, alphabets, and labeled Petri nets, which renames appropriately its operand by priming unobservable events and the label of each place. We also change the labels of every place in the supervision patterns Ω_i from N (reps. F) to N_i (resp. F_i).

We start by combining the set of supervision patterns in a meta-pattern $\Pi = (\Omega_1 \parallel \Omega_2 \parallel \ldots \parallel \Omega_n)$ — the product is obviously associative — then we perform the product of Θ and Π to obtain $\Theta_\Pi = \Theta \star \Pi$. The net Γ is calculated as the synchronized product of Θ_Π and Θ'_Π — the result of this product is commonly called the twin-plant. The final step consists in finding the possible sequences $c : M_{\Gamma_0} \xrightarrow{c} M$ from M_{Γ_0} the initial marking of Γ that lead to an ambiguous marking that can indefinitely occur.

Definition 13. A marking M in Γ is ambiguous if: $\exists p, p' \in P_\Gamma, M(p) \times M(p') \neq 0 \wedge ((\ell(p) = F_i \wedge \ell(p') = N'_i) \vee (\ell(p) = N_i \wedge \ell(p') = F'_i))$.

Definition 14. A set of supervision patterns $\mathbf{\Omega}^s$ is said to be *involved* in an ambiguous marking M, if:
$$F_i \in \ell(M) \iff \Omega_i \in \mathbf{\Omega}^s$$

Definition 15. A sequence of transitions $c \in T^*_\Gamma$ is ambiguous if it leads to an ambiguous marking.

Definition 16. A set of supervision patterns $\mathbf{\Omega}^s$ is said to be involved in sequence of transitions $c \in T^*_\Gamma$ leading to a marking M, if $\mathbf{\Omega}^s$ is involved in M.

We set $Q_\Gamma = \{q = q_1 \cup q_2 : q_1 \in \ell^{-1}_{\Theta_\Pi}(F_i), q_2 \in \ell^{-1}_{\Theta'_\Pi}(N'_i), \text{ for } i = 1..n\}$.

Proposition 17. A reachable marking M of Γ is ambiguous $\iff \exists c \in T^*_\Gamma : M_{\Gamma_0} \xrightarrow{c} M$ and $\ell_\Gamma(c) \in \mathcal{L}(\Gamma)$.

Definition 18. A sequence of transitions $c \in T^*_\Gamma$ is an admissible cycle in Γ if $\exists s \in T^*_\Gamma, \exists t \in T^+_\Gamma : \forall n \in \mathbb{N}, \ell_\Gamma(st^n) \in \mathcal{L}(\Gamma) \wedge c = st$.

From Definitions 15, 18 and Proposition 17

Proposition 19. c is an admissible cycle in $\Gamma \implies c$ is an ambiguous cycle.

The diagnosability analysis (generalization of Gougam et al. (2013)) relies on the following result.

Theorem 20. Θ is not $\mathbf{\Omega}$-diagnosable $\iff \Gamma$ contains ambiguous cycles.

The notion of discriminability proposed in this paper can be analyzed based on the following:

Theorem 21. A subset $\mathbf{\Omega}^s \in 2^{\mathbf{\Omega}}$ of supervision patterns is said to be discriminable $\iff \mathbf{\Omega}^s$ is not involved in any ambiguous cycle of Γ.

So, the problem of patterns discriminability and $\mathbf{\Omega}$-diagnosability can be reduced to finding ambiguous cycles in Γ.

To sum up, the method includes:

(1) modelling the supervision patterns and the system with labeled Petri nets;
(2) compute a meta-pattern and combine it with the system;
(3) construction of a twin-plant;
(4) unfolding the labeled Petri net of the twin-plant;
(5) search for ambiguous cycles in the twin-plant;
(6) conclude on the discriminability of the patterns and consequently on the system diagnosability.

5. FINDING AMBIGUOUS CYCLES

To find cycles in a Petri net, we will use *Petri net unfoldings*. This technique — introduced by McMillan (1995) — attempts to avoid the combinatorial explosion inherent to the classical usage of *marking graphs*.

5.1 Petri net unfolding

A Petri net unfolding is another labeled Petri net that is generally infinite.

Definition 22. The unfolding $\Phi = \langle P_\Phi, T_\Phi, A_\Phi, \ell_\Phi, P, T, M_{\Phi_0}\rangle$ of a marked and labeled Petri net $\Theta = \langle P, T, A, \ell, L, \Sigma, M_0\rangle$, is a labeled Petri net such that:

(1) $\forall p \in P_\Phi : |\bullet p| \leq 1$.
(2) Φ is acyclic, i.e. for any element $x \in P_\Phi \cup T_\Phi : \neg(x < x)$
(3) Φ is finitely preceded, i.e., for every $x \in P_\Phi \cup T_\Phi$, the set of elements $y \in P_\Phi \cup T_\Phi$ such that $y < x$ is finite.
(4) No $x \in P_\Phi \cup T_\Phi$ is in self-conflict $\neg(x \# x)$.
(5) $\ell_\Phi(P_\Phi) \subseteq P$, $\ell_\Phi(T_\Phi) \subseteq T$;
(6) $\forall t \in T_\Phi : \bullet\ell_\Phi(t)$ is isomorphic to $\bullet t$;
(7) $\forall t \in T_\Phi : \ell_\Phi(t)\bullet$ is isomorphic to $t\bullet$;
(8) $M_{\Phi_0} = \{x \in P_\Phi : \nexists y \in T_\Phi, y < x\}$.

Where $<$ is the transitive closure of the arc relation A_Φ — called the causal relation, and $\leq$ the reflexive closure of $<$. The couple of elements $(x, y) \in (P_\Phi \cup T_\Phi)^2$ is a conflict (denoted $x \# y$) if there exist two transitions $t_1, t_2 \in T_\Phi$, such that $t_1 \neq t_2$, $\bullet t_1 \cup \bullet t_2 \neq \varnothing$ and $(t_1, x), (t_2, y) \in <$.

A *configuration* C of an unfolding is a set of causally-closed and conflict-free transitions of T_Φ, formally: $t \in C \Rightarrow \forall t' < t, t' \in C$ and $\forall t, t' \in C : \neg(t \# t')$. The cut of a configuration C is the set of places $\mathrm{Cut}(C) = (M_{\Phi_0} \cup C\bullet) \setminus \bullet C$ where $C\bullet = \{p \in t\bullet, t \in C\}$ and $\bullet C = \{p \in \bullet t, t \in C\}$ and $\ell_\Phi(\mathrm{Cut}(C))$ characterizes a marking of the net Θ. The *local configuration* of a transition $t \in T_\Phi$ is the set of transitions $t' \in T_\Phi$ such that $t' \leq t$.

For any bounded Petri net, McMillan (1995) shows the existence of a fragment of the unfolding Φ also called the *finite complete prefix* such that any reachable marking of Θ is represented by the cut of a configuration of that fragment. The construction of such a prefix relies on the search for *cut-off points* (McMillan (1995), Esparza et al. (2002)). A cut-off point is a transition $t_c \in T_\Phi$ such that its local configuration $[t_c]$ contains a transition t' such that $\ell_\Phi(\mathrm{Cut}([t'])) = \ell_\Phi(\mathrm{Cut}([t_c]))$ and the relation $[t'] \prec [t]$ holds, where $\prec$ is an *adequate order* (see Esparza et al. (2002)). Let T_c be the set of cut-off events which local configuration contains exactly one cut-off event. The finite complete prefix of Φ is then defined as $\bigcup_{t \in T_c}\{[t] \cup t\bullet\}$. For every adequate order there is a different finite complete prefix of Φ appropriate to analyze a particular property of the Petri net. A partial order which leads to a complete finite prefix appropriate for detecting cycles is the strict set inclusion "$\subset$".

5.2 Algorithm

Algorithm 1 looks for ambiguous cycles in the twin-plant Γ.

Algorithm 1 Search for ambiguous cycles

1: **procedure** ANALYSIS(Γ)
2: $\quad\Phi \leftarrow$ unfolding(Γ)
3: $\quad H \leftarrow$ cut-offs(Φ)
4: $\quad$**for all** e in $H.events$ **do**
5: $\qquad$**if** $\ell_\Phi(H[e])$ is ambiguous **then**
6: $\qquad\quad$prune($H[e]$)
7: $\qquad$**end if**
8: $\quad$**end for**
9: **end procedure**

line 2: compute the unfolding of the Petri net Γ.
line 3: build a mapping between the cut-off events and their associated markings. This step can be performed while computing the unfolding by finding the maximal cuts — which represent reachable markings — including the postset of every cutoff event.
line 4–8: prune from H every non-ambiguous marking.

Once the algorithm is applied, we check H for possible remaining markings. If there is any, every supervision pattern involved in any marking is not discriminable and the system is not $\mathbf{\Omega}$-diagnosable. Otherwise, the supervision patterns are discriminable and the system is $\mathbf{\Omega}$-diagnosable.

The following section illustrates the method with the help of an example.

6. EXAMPLE

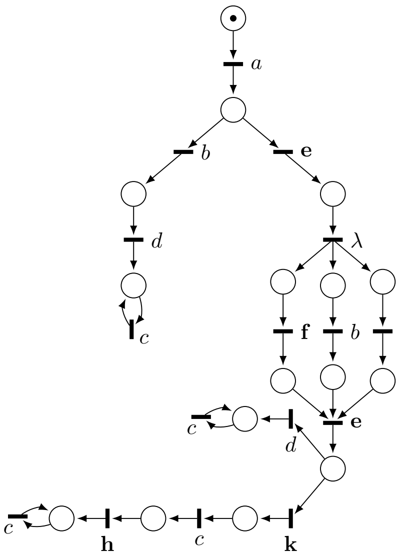
The method previously presented was implemented and in order to illustrate it, we present hereafter an example. The system is modeled by Θ, $\Sigma_{uo} = \{e, f, g, k, h\}$ represented in bold in Figure 2. There is two patterns Ω_1 and Ω_2, the first models the parallel occurrence of two events $\{b, g\}$, the second models the sequential occurrence of k and h.

The table 1 summarizes the size of the Petri nets obtained at each step.

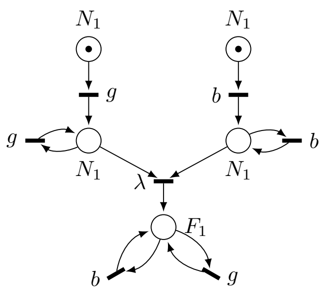
Table 1. Size of intermediary Petri nets of the method

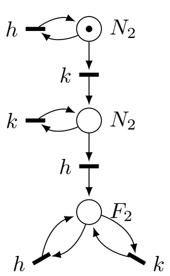
	$\|P\|$	$\|T\|$
System (Θ)	16	16
First pattern (Ω_1)	5	7
Second pattern (Ω_2)	3	6
Meta-pattern (Π)	8	13
Product of the system and the meta-pattern (Θ_Π)	24	27
Twin-plant (Γ)	48	85
Unfolding of the twin-plant (Φ)	82	38

We can see that even though the example is not strongly concurrent, the overhead in places and transitions is



(a) The System Θ





(b) Pattern Ω_1 (c) Pattern Ω_2

Figure 2. Modeling of the system and the pattern

reasonable. Plus, the completeness condition imposed on supervision pattern results in extra transitions that are striped away by the unfolding, whence the reduction of the number of transitions from Γ to Φ.

Finally, the remaining markings in H will have the following labels: $\{-, N_1, N_2, F_1', N_2'\}$ and $\{-, N_1', N_2', F_1, N_2\}$. This means that Ω_1 is not discriminable, and consequently that Ω_2 and the combination $\Omega_1 \& \Omega_2$ are discriminable.

7. CONCLUSION AND PERSPECTIVES

In this paper, we introduced the notion of discriminability of supervision patterns and proposed a method to analyse it. The discriminability of a pattern ensures that one can detect the exclusive occurrence of this particular pattern, which differs from its mere recognition without further information about other patterns. We have also shown that once the discriminability analysis is performed, we can conclude about the diagnosability of the system with no extra cost. Our approach relies on the construction of a meta-pattern embedding all interesting behaviors. After combining it with the system, we perform a twin-plant to detect ambiguous sequences by using petri net unfoldings.

An interesting continuation of this work could be the extension of the models to include timed systems and patterns. This would permit us to monitor more complex — and interesting — behaviors that can not be modeled otherwise.

REFERENCES

Basile, F., Chiacchio, P., and De Tommasi, G. (2009). An efficient approach for online diagnosis of discrete event systems. *Transactions on Automatic Control*, 54(4), 748–759.

Benveniste, A., Fabre, E., Haar, S., and Jard, C. (2003). Diagnosis of asynchronous discrete-event systems: a net unfolding approach. *Transactions on Automatic Control*, 48(5), 714–727.

Cabasino, M.P., Giua, A., Lafortune, S., and Seatzu, C. (2012). A new approach for diagnosability analysis of petri nets using verifier nets. *Transactions on Automatic Control*, 57(12), 3104–3117.

Dotoli, M., Fianti, M.P., Mangini, A.M., and Ukovich, W. (2009). On-line fault detection in discrete event systems by petri nets and integer linear programming. *Automatica*, 45(11), 2665–2672.

Esparza, J., Römer, S., and Vogler, W. (2002). An Improvement of McMillan's Unfolding Algorithm. *Formal Methods in System Design*, 20(3), 285–310.

Gougam, H.E., Subias, A., and Pencolé, Y. (2013). Supervision patterns: Formal diagnosability checking by petri net unfolding. In *Dependable Control of Discrete Systems*, volume 4.

Jéron, T., Marchand, H., Pinchinat, S., and Cordier, M.O. (2006). Supervision Patterns in Discrete Event Systems Diagnosis. In *Workshop on Discrete Event Systems, WODES'06*, 262–268. IEEE Computer society, Ann-Arbor, USA.

Jiang, S., Huang, Z., Chandra, V., and Kumar, R. (2000). A polynomial algorithm for testing diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*.

Jiang, S., Kumar, R., and Garcia, H.E. (2003). Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Transactions on Robotics*, 19(2), 310–323.

Lai, S., Nessi, D., Cabasino, M., Giua, A., and Seatzu, C. (2008). A comparison between two diagnostic tools based on automata and petri nets. In *9th International Workshop on Discrete Event Systems*, 144–149.

McMillan, K.L. (1995). A technique of state space search based on unfolding. *Formal Methods in System Design*, 6(1), 45–65.

Pencolé, Y. (2005). Assistance for the design of a diagnosable component-based system. In *17th International Conference on Tools with Artificial Intelligence (ICTAI 05)*, 549–556. Hong-Kong, China.

Peterson, J.L. (1977). Petri nets. *ACM Comput. Surv.*, 9(3), 223–252.

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

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

Yoo, T.S. and Lafortune, S. (2002). Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Trans. Automat. Contr.*, 47(9), 1491–1495.