Diagnosticabilité de motifs de supervision par dépliage de réseaux de Petri

Houssam-Eddine GOUGAM[1,2], Audine SUBIAS[1,2], Yannick PENCOLÉ[1,3]

[1] CNRS ; LAAS ; 7, avenue du Colonel Roche, F-31400 Toulouse, France

[2] Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France

[3] Univ de Toulouse, LAAS, F-31400 Toulouse, France

{gougam,subias,pencole}@laas.fr

Résumé— **Ce papier s'intéresse au problème de la diagnosticabilité des motifs de supervision dans les systèmes à événements discrets. Ces motifs permettent de prendre en compte des comportements de fautes complexes. La méthode d'analyse de diagnosticabilité s'appuie sur l'utilisation des réseaux de Petri et sur les méthodes de dépliage de manière à appréhender les problèmes d'explosion combinatoire induits par des approches classiques d'analyse par graphe des marquages. L'approche proposée consiste à vérifier la diagnosticabilité de chaque motif en adaptant la méthode classique du produit jumelé (*twin-plant*). La non-diagnosticabilité du motif de faute est vérifiée par la recherche de séquences infinies ambiguës dans le dépliage du produit.**

Mots-clés— **diagnosticabilité, motifs de supervision, réseaux de Petri étiquetés, dépliage.**

I. Introduction

Surveiller un système, un procédé en vue de maintenir ses performances dans le temps est une tâche critique qui nécessite le déploiement d'un ensemble d'outils de supervision (capteurs, moniteurs, diagnostiqueurs, etc). La tâche essentielle du superviseur est de reconnaître avec le plus de certitude possible une situation critique (citons par exemple la présence d'un défaut, d'une faute, d'une panne, un état de fonctionnement non souhaité, à risque ou non sûr). Dans le cadre des systèmes à événements discrets (SED), la fonction de supervision consiste à analyser la séquence d'événements observée et à déterminer si une telle situation a pu se produire. Par la nature même des SED, une situation est caractérisée par un agencement d'événements qui peuvent être observables ou non, nous les appellerons ici des *motifs* d'événements.

Dans cet article, nous nous intéressons au problème de la diagnosticabilité de tels motifs, c.-à-d. à l'analyse de la capacité de la fonction de supervision à déterminer, à diagnostiquer *avec certitude* la présence d'un motif particulier à partir d'une séquence d'événements observés [1]. Ce problème n'est pas original et de nombreux travaux ont déjà été développés, proposant de nombreuses techniques d'analyses formelles [2], [3] ; mais la difficulté réside dans l'explosion combinatoire intrinsèque de cette analyse. Notre originalité sur le sujet est double. Premièrement le SED et les motifs à surveiller sont représentés à l'aide de réseau de Petri [4], [5], [6] ce qui permet de représenter facilement la nature généralement distribuée des systèmes mais également la concurrence d'événements dans les systèmes et dans les motifs. Deuxièmement, afin d'améliorer l'efficacité de l'analyse, nous proposons une approche par dépliage [7], [8] afin de tirer partie d'une représentation par ordre partielle qui évite l'énumération des séquences d'événements qui se produit notamment si l'analyse se fait par la construction d'un graphe de marquage [9].

L'article est organisé de la façon suivante. Après un rappel des notions formelles utilisée tout au long de cet article en section II, la section III introduit les motifs de supervision représentés par réseau de Petri. La section V décrit la méthode d'analyse proposée qui est entièrement illustrée par un exemple en section VI.

II. Notions de base

Cette partie rappelle brièvement les fondements théoriques sur lesquels nous avons développé nos travaux.

A. Réseaux de Petri étiquetés

Définition II.1 : Un réseau de Petri étiqueté (RdPÉ) et marqué est un 8-uplet $\langle P, T, Pre, Post, \ell, L, \Sigma, m_0 \rangle$, avec :

- P : un ensemble fini de places ;
- T : un ensemble fini de transitions, avec $P \cap T = \varnothing$;
- $Pre : P \times T \to \mathbb{N}$: l'application places précédentes ;
- $Post : P \times T \to \mathbb{N}$: l'application places suivantes ;
- $\ell : P \cup T \to L \cup \Sigma \cup \{\lambda\}$: l'application d'étiquetage, où L est l'ensemble des étiquettes de place, Σ l'ensemble des étiquettes de transition et λ la séquence vide ;
- $m_0 : P \to \mathbb{N}$ l'application marquage initial, qui associe à chaque place le nombre de jetons qu'elle contient initialement.

L'état courant du réseau de Petri est défini par un marquage $m : P \to \mathbb{N}$ qui associe à chaque place le nombre de jetons qu'elle contient, l'état initial étant caractérisé par m_0. Le franchissement de transitions permet de définir les successeurs d'un état. Une transition du réseau de Petri est franchissable à partir d'un marquage donné m, ssi : $\forall p \in P, m(p) \geq Pre(p,t)$. Si le franchissement d'une transition t à partir du marquage m conduit au marquage m', on notera : $m \xrightarrow{t} m'$. Cette notation se généralise aussi à une séquence de transition $s = t_1 t_2 \dots t_n$.

L'analyse de réseaux de Petri peut être réalisée au travers du graphe des marquages. Cette technique présente toutefois l'inconvénient de devoir énumérer toutes les évolutions parallèles, ce qui conduit généralement à l'explosion combinatoire du nombre d'états. Pour remédier à ce problème,

[7] propose d'utiliser le *dépliage de réseau de Petri* qui est une technique d'ordre partiel.

B. Dépliage de réseaux de Petri

Le dépliage de réseau de Petri est un autre réseau de Petri étiqueté — généralement infini — avec une structure plus simple que celle du réseau initial.

Définition II.2 : Le dépliage $\mathcal{D} = \langle P_\mathcal{D}, T_\mathcal{D}, Pre_\mathcal{D}, Post_\mathcal{D}, \ell_\mathcal{D}, P, T, m_{\mathcal{D}_0}\rangle$, d'un réseau de Petri $\mathcal{R} = \langle P, T, Pre, Post, \ell, L, \Sigma, m_0\rangle$, est un réseau de Petri qui vérifie les propriétés suivantes.

1. $\forall p \in P_\mathcal{D} : \sum_{t\in T_\mathcal{D}} Pre(p,t) \leq 1$.
2. $\mathcal{D}$ est acyclique.
3. $\mathcal{D}$ est fini par précédence.
4. Pour tout nœud x de $\mathcal{D}$, c.-à-d. $x \in P_\mathcal{D} \cup T_\mathcal{D}$, x n'est pas en conflit avec lui-même. Un nœud est dit en conflit avec lui-même s'il existe, à partir d'une place, deux chemins différents permettant de l'atteindre.

L'application d'étiquetage, doit aussi vérifier les propriétés suivantes :

1. $\ell_\mathcal{D}(P_\mathcal{D}) \subset P$, $\ell_\mathcal{D}(T_\mathcal{D}) \subset T$;
2. $\forall t \in T_\mathcal{D} : \bullet\ell_\mathcal{D}(t)$ est une bijection vers $\ell(\bullet t)$;
3. $\forall t \in T_\mathcal{D} : \ell_\mathcal{D}(t)\bullet$ est une bijection vers $\ell(t\bullet)$.

avec, pour un RdPÉ quelconque $\mathcal{R} = \langle P, T, Pre, Post, \ell, L, \Sigma, m_0\rangle$: $\bullet t = \{p \in P : Pre(p,t) > 0\}$, et $t\bullet = \{p \in P : Post(p,t) > 0\}$.

Le dépliage pouvant être infini, l'analyse ne se fait pas directement dessus, mais sur une sous partie appelée *préfixe complet*. Cette sous partie contient tous les marquages accessibles du réseau de Petri, au moins une fois.

La construction du préfixe complet se base sur le concept de *transitions de coupe (cut-off)*. Une transition t_c du dépliage est dite de coupe si son franchissement mène vers le même marquage que le franchissement d'une autre transition t et la longueur de la trace contenant t_c est strictement supérieure à celle de la trace contenant t, c.-à-d. le franchissement de t_c mène vers un marquage déjà rencontré. Enfin, le préfixe complet est construit à partir du dépliage en ne gardant que les nœuds — places et transitions — précédant les transitions de coupe. L'existence et la construction d'un tel préfixe est décrite plus en détail dans [7] et [8].

Dans la suite, on utilisera indifféremment les termes préfixe complet et dépliage pour parler du préfixe complet.

C. Produit synchronisé de réseaux de Petri étiquetés

Le produit synchronisé de deux RdPÉ est un troisième RdPÉ représentant l'intersection de leurs comportements.

Soient deux RdPÉ, $\mathcal{N}_1 = \langle P_1, T_1, Pre_1, Post_1, \ell_1, L_1, \Sigma_1, m_{10}\rangle$ et $\mathcal{N}_2 = \langle P_2, T_2, Pre_2, Post_2, \ell_2, L_2, \Sigma_2, m_{20}\rangle$, leur produit synchronisé sur un ensemble d'étiquettes de transitions Σ_s, $\mathcal{N} = \mathcal{N}_1 \|_{\Sigma_s} \mathcal{N}_2$, est le RdPÉ $\langle P, T, Pre, Post, \ell, L, \Sigma, m_0\rangle$ défini comme suit :

- $P = P_1 \cup P_2$;
- $T = \bigcup_{l\in\Sigma_s} \{t_1\|t_2 | \exists (t_1,t_2) \in (T_1 \times T_2) : \ell_1(t_1) = \ell_2(t_2) = l\} \cup \{t \in T_1 | \ell_1(t) \notin \Sigma_s\} \cup \{t \in T_2 | \ell_2(t) \notin \Sigma_s\}$;
- $Pre(p,t) = \begin{cases} Pre_1(p,t) & \text{si } (p,t) \in P_1 \times T_1 \\ Pre_2(p,t) & \text{si } (p,t) \in P_2 \times T_2 \\ Pre_1(p,t_1) & \text{si } p \in P_1 \wedge t = t_1\|t_2 \\ Pre_2(p,t_2) & \text{si } p \in P_2 \wedge t = t_1\|t_2 \\ 0 & \text{sinon} \end{cases}$;
- $Post(p,t) = \begin{cases} Post_1(p,t) & \text{si } (p,t) \in P_1 \times T_1 \\ Post_2(p,t) & \text{si } (p,t) \in P_2 \times T_2 \\ Post_1(p,t_1) & \text{si } p \in P_1 \wedge t = t_1\|t_2 \\ Post_2(p,t_2) & \text{si } p \in P_2 \wedge t = t_1\|t_2 \\ 0 & \text{sinon} \end{cases}$;
- $\ell(p) = \begin{cases} \ell_1(p) & \text{si } p \in P_1 \\ \ell_2(p) & \text{si } p \in P_2 \end{cases}$;
- $\ell(t) = \begin{cases} \ell_1(t_1) & \text{si } t = t_1\|t_2 \\ \ell_1(t) & \text{si } t \in T_1 \\ \ell_2(t) & \text{si } t \in T_2 \end{cases}$;
- $L = L_1 \cup L_2$;
- $\Sigma = \Sigma_1 \cup \Sigma_2$;
- $m_0(p) = \begin{cases} m_{10}(p) & \text{si } p \in P_1 \\ m_{20}(p) & \text{si } p \in P_2 \end{cases}$.

Ce produit généralise celui défini dans [10]. La différence réside dans le fait que ce produit peut se faire sur un ensemble d'étiquettes quelconque, alors que celui de [10] se fait nécessairement sur l'ensemble des étiquettes communes de $\mathcal{N}_1$ et $\mathcal{N}_2$. Autrement dit, le produit de [10] est le produit $\mathcal{N} = \mathcal{N}_1 \|_{\Sigma_1\cap\Sigma_2} \mathcal{N}_2$.

III. Motifs de supervision

Le problème de la détection et du diagnostic de faute dans les systèmes à événements discrets s'appuie généralement sur un modèle de comportement fautif représenté par l'occurrence d'un événement particulier — représentant une faute [1]. Les motifs de supervision étendent ce type de modèle en y introduisant des comportements plus complexes [2], [3], impliquant un ou plusieurs événements et dont seul l'agencement décrit dans le motif est considéré comme fautif. Nous proposons ici de représenter ces motifs à l'aide de RDPÉ.

Définition III.1 : Un *motif de supervision* est un comportement du système qui peut être modélisé par un réseau de Petri étiqueté et marqué $\langle P, T, Pre, Post, \ell, L, \Sigma, m_0\rangle$ ayant la structure suivante :

1. $L = \{N, F\}$, les places sont étiquetées par N si elles correspondent à un état dit *normal* du système (le motif n'a pas eu lieu) et par F si elles correspondent à un état dit *fautif* (le motif a eu lieu) ;
2. $\forall p \in P : m_0(p) > 0 \Rightarrow \ell(p) = N$, c.-à-d. le marquage initial ne contient aucune place fautive ;
3. $\forall p_1, p_2 \in P : \ell(p_1) = N \wedge \ell(p_2) = F \Rightarrow m(p_1) \times m(p_2) = 0$, c.-à-d. deux places, l'une normale et l'autre fautive, ne peuvent être marquées en même temps ;
4. $\forall m : (\exists s = t_1 t_2 \ldots : m_0 \xrightarrow{s} m), \forall e \in \Sigma, \exists t \in T : (\ell(t) = e \wedge \forall p \in P, m(p) \geq Pre(p,t))$, c.-à-d. le motif doit pouvoir évoluer — depuis un marquage atteignable à partir de m_0 — à chaque occurrence d'un événement de son alphabet Σ ;
5. $\forall m : (\exists s = t_1 t_2 \ldots t_n : m_0 \xrightarrow{s} m) : (m' \xrightarrow{t_n} m \wedge \exists p \in P : m'(p) > 0 \wedge \ell(p) = F \Rightarrow \exists p \in P : m(p) = F)$, c.-à-d. toute

évolution du motif depuis un état fautif, conduit vers un état fautif.

A. Exemple de motif de supervision

Nous considérons tout type de motifs de supervision issus de la définition III.1, sachant que certains motifs sont plus pertinents que d'autres. Parmi les plus intéressants, on a notamment ceux proposés par [3] qui sont tous traduisibles en motif sous forme de RdPÉ. Nous présentons ici, à titre d'exemples, la traduction de deux d'entre eux.

A.1 Motif de l'occurrence de plusieurs événements en parallèle

Soit $E = \{e_1, e_2, \ldots\}$, un ensemble d'événements. Le motif correspondant à l'occurrence de tous les éléments de E dans un ordre quelconque, est modélisé par le RdPÉ $\langle P, T, Pre, Post, \ell, L, E, m_0\rangle$ défini comme suit :

- $P = \{p_0\} \cup \bigcup_{i=1}^{2} \bigcup_{j=1}^{|E|} \{p_{ij}\}$;
- $T = \{t_0\} \cup \bigcup_{i=1}^{3} \bigcup_{j=1}^{|E|} \{t_{ij}\}$;
- $Pre(p,t) = \begin{cases} 1 & \text{si } \substack{(p,t)\in\bigcup_{i=1}^{|E|}\{(p_{1i},t_{1i}),(p_{2i},t_{2i}),(p_{2i},t_0),\\ (p_0,t_{3i})\}} \\ 0 & \text{sinon} \end{cases}$;
- $Post(p,t) = \begin{cases} 1 & \text{si } \substack{(p,t)\in\bigcup_{i=1}^{|E|}\{(p_{2i},t_{1i}),(p_{2i},t_{2i}),(p_0,t_{3i})\}\cup\\ \{(p_0,t_0)\}} \\ 0 & \text{sinon} \end{cases}$;
- $\ell(p) = \begin{cases} F & \text{si } p = p_0 \\ N & \text{sinon} \end{cases}$;
- $\ell(t) = \begin{cases} \lambda & \text{si } t = t_0 \\ e_j & \text{si } \substack{t=t_{ij} \text{ pour } i\in\{1,2,3\} \text{ et}\\ j\in\{1,\ldots,|E|\}} \end{cases}$;
- $m_0(p) = \begin{cases} 1 & \text{si } p \in \bigcup_{i=1}^{|E|}\{p_{1i}\} \\ 0 & \text{sinon} \end{cases}$.

La figure 1 donne une représentation graphique du réseau.

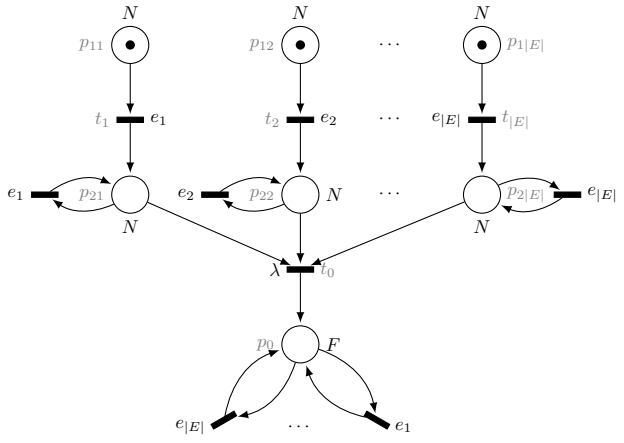

Fig. 1. Motif modélisant l'occurrence de plusieurs événements en parallèle

Dans toutes les figures représentant des réseaux de Petri étiquetés, le texte en gris resprésente les noms des places et des transitions, et le texte en noir est utilisé pour leurs étiquettes.

La transition t_0 sert à vérifier que tous les événements de E ont eu lieu ; elle est étiquetée par λ, ce qui la rend franchissable après l'occurrence de tous les événements du motif.

Cet exemple illustre parfaitement l'intérêt de l'utilisation des réseaux de Petri au lieu des automates. Là où le motif est de taille $2^{|E|}$ avec une modélisation par automates [3], sa taille n'est que de $2|E| + 1$ dans le cas des réseaux de Petri.

B. Motif de l'occurrence multiple du même événement

Le motif de supervision correspondant à l'occurrence d'un événement e, k fois ($k > 0$), est modélisé par le RdPÉ $\langle P, T, Pre, Post, \ell, L, E, m_0\rangle$ défini dans la suite.

- $P = \bigcup_{i=0}^{k} \{p_i\}$;
- $T = \bigcup_{i=0}^{k} \{t_i\}$;
- $Pre(p,t) = \begin{cases} 1 & \text{si } (p,t) \in \bigcup_{i=0}^{k} \{(p_i, t_i)\} \\ 0 & \text{sinon} \end{cases}$;
- $Post(p,t) = \begin{cases} 1 & \text{si } (p,t) \in \bigcup_{i=0}^{k-1} \{(p_{i+1}, t_i)\} \cup \{(p_k, t_k)\} \\ 0 & \text{sinon} \end{cases}$;
- $\ell(p) = \begin{cases} F & \text{si } p = p_k \\ N & \text{sinon} \end{cases}$;
- $\ell(t) = e, \forall t \in T$;
- $\Sigma = \{e\}$;
- $m_0(p) = \begin{cases} 1 & \text{si } p = p_0 \\ 0 & \text{sinon} \end{cases}$.

La figure 2 représente le RdPÉ de manière graphique.

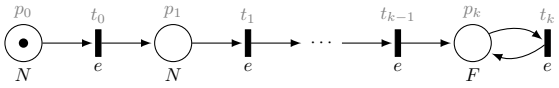

Fig. 2. Motif modélisant l'occurrence, k fois, d'un événement e

Ce motif ressemble structurellement au motif modélisant k occurrences du même événement, défini dans [3]. Cette ressemblance n'est pas fortuite, puisque la construction des autres motifs se fait de la même manière. Cela est dû au fait que ces motifs n'ont pas de composante concurrentielle. On remarquera toutefois que ce type de motifs définis dans le cadre des réseaux de Petri est plus concis que ceux de [3] car ils ne prennent en compte que les événements $\Sigma = \{e\}$ du motif et non pas tous les événements du système.

Enfin, ce motif peut trivialement être utilisé pour modéliser l'occurrence d'une faute simple, il suffit de prendre $k = 1$ et l'on retrouve ainsi le cadre classique du problème de diagnosticabilité dans les SED [1], [9].

IV. Diagnosticabilité des motifs de supervision

Le problème de la diagnosticabilité des motifs de supervision dans un système est défini comme suit. Le système à analyser est modélisé par un réseau de Petri étiqueté et marqué $\mathcal{N} = \langle P, T, Pre, Post, \ell, L, \Sigma, m_0\rangle$. L'alphabet Σ

représente l'ensemble des événements que le système génère, et peut être divisé en deux sous ensembles : Σ_o et Σ_u, représentant respectivement l'ensemble des événements observables et l'ensemble des événements non-observables. La figure 4 décrit le modèle d'un tel système.

Le motif de supervision à analyser est modélisé par le RdPÉ $\Omega = \langle P_\Omega, T_\Omega, Pre_\Omega, Post_\Omega, \ell_\Omega, L_\Omega, \Sigma_\Omega, m_{\Omega 0}\rangle$, où généralement $\Sigma_\Omega \subset \Sigma$ (voir par exemple la figure 5 illustrant un motif de supervision pour le système de la figure 4).

Dans cet article, l'analyse que nous proposons consiste à déterminer si le système est Ω-*diagnosticable* au sens de [3] — avec Ω un motif de supervision quelconque. Autrement dit, on cherche à déterminer s'il suffit toujours d'attendre un nombre *fini* d'observations, dès lors que le motif Ω a eu lieu, pour diagnostiquer *sans ambiguïté* qu'il a effectivement eu lieu et qu'il y a donc pas d'autres explications possibles selon le modèle $\mathcal{N}$.

De manière générale, la méthode classique pour analyser la diagnosticabilité de fautes simples dans un système, introduite séparément dans [11] et [12], consiste à réaliser le produit jumelé du système sur l'ensemble des observables. L'idée étant que ce produit permet de trouver les traces normales et fautives qui ont le même comportement observable — si elles existent — et ainsi de conclure sur la diagnosticabilité ou non du système en cherchant parmi celles-ci, celles qui ont un nombre infini d'événements observables.

Dans le cadre des motifs de supervision, le même raisonnement s'applique. Cependant, avant de pouvoir utiliser le produit jumelé, il faut tenir compte du comportement du motif de supervision. Cela est réalisé avec le produit synchronisé du système avec le motif de supervision.

Les étapes de l'analyse de la diagnosticabilité sont présentées plus en détail, dans le section suivante.

V. Analyse de diagnosticabilité

A. Étapes de l'analyse

Le diagramme de la figure 3 illustre les étapes nécessaires pour l'étude de la diagnosticabilité. La méthode que nous proposons est une adaptation de celle de [3] aux réseaux de Petri étiquetés. La principale différence étant l'utilisation d'un produit *synchronisé* au lieu du produit *synchrone* et le recours aux dépliage des réseaux de Petri pour l'analyse.

À noter que les langages des réseaux de Petri modélisant les motifs de supervision ne sont pas nécessairement universels, contrairement à ceux des automates de [3]. Ceci est une conséquence de l'utilisation du produit synchronisé à la place d'un produit synchrone.

Dans un premier temps, le produit synchronisé de $\mathcal{N}$et de Ω est effectué sur l'ensemble des étiquettes de transitions donné par : $\Sigma_s = \Sigma \cap \Sigma_\Omega$, pour obtenir le réseau noté $\mathcal{N}_\Omega$ (voir sous-section II-C).

Ensuite, $\mathcal{N}'_\Omega$ est défini comme étant le réseau de Petri $\mathcal{N}_\Omega$, où les places, transitions et étiquettes de place sont renommées en y ajoutant un «$'$». Le produit synchronisé de $\mathcal{N}_\Omega$ et de $\mathcal{N}'_\Omega$ sur Σ_o, qui est communément appelé le produit jumelé (*twin-plant*), est noté Γ.

Enfin, l'analyse à proprement parler consiste à trouver, dans Γ, les traces franchissables — à partir de m_0 — menant à un marquage ambigu et qui peuvent durer indéfiniment.

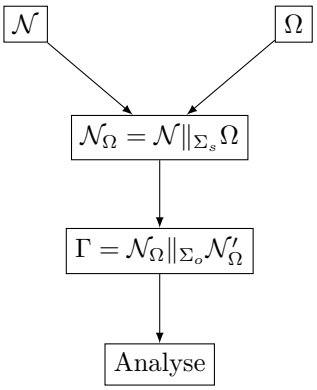

Fig. 3. Les étapes de l'analyse de diagnosticabilité

Un marquage m dans Γ est dit ambigu ssi : $\exists p, p' \in P_\Gamma, m(p) \times m(p') \neq 0 \wedge \ell(p) = N \wedge \ell(p') = F'$.

La section suivante détaille cette étape de l'analyse.

B. Algorithme d'analyse

La recherche des marquages ambigus se fait selon l'algorithme 1.

Algorithme 1 Recherche des traces ambiguës

```
1 :  diagnosticabilité : fonction ANALYSE(RdPE)
2 :      ⟨D, H⟩ ← dépliage(RdPE)
3 :      pour tout t dans H.transitions faire
4 :          si {N', F} ⊄ ℓ(ℓ_D(H[t]))∧{N, F'} ⊄ ℓ(ℓ_D(H[t]))
     alors
5 :              effacer(H[t])
6 :          fin si
7 :      fin pour
8 :      si H est vide alors
9 :          retourne « diagnosticable »
10 :     sinon
11 :         retourne « non-diagnosticable »
12 :     fin si
13 : fin fonction
```

ligne 4 : la fonction *dépliage* prend un réseau de Petri étiqueté en entrée et retourne en sortie son dépliage $\mathcal{D}$ et une structure H contenant les transitions de coupe (*cut-off*) avec le marquage résultant de leur franchissement. Les transitions de coupe correspondent aux cycles dans le réseau de Petri (voir sous-section II-B).

lignes 5-9 : tous les éléments de la structure H qui ne mènent pas vers un marquage ambigu sont supprimés.

lignes 10-11 : si H est vide, alors il n'y a pas de marquage ambigu qui dure indéfiniment. Donc, le motif est diagnosticable.

lignes 12-14 : si H n'est pas vide, c.-à-d. il existe — dans le dépliage — au moins une trace contenant une transition de coupe et menant à un marquage ambigu, ce qui revient à dire que le réseau de Petri peut rester dans un état ambigu indéfiniement, le système n'est pas diagnosticable.

La section suivante illustre l'approche proposée de diagnosticabilité de motifs de supervision par dépliage de réseaux de Petri.

VI. Exemple

Pour pouvoir illustrer toutes les étapes de la méthode, un exemple très simple a été choisi. L'étude de la diagnosticabilité du système $\mathcal{N}$ représenté pas la figure 4, vis-à-vis du motif de supervision Ω — figure 5 — représentant l'occurrence d'un événement de faute simple.

Tous les événements du système $\mathcal{N}$ sont observables sauf u et f. Donc, $\Sigma = \{a, b, u, f\}$, $\Sigma_o = \{a, b\}$ et $\Sigma_u = \{u, f\}$.

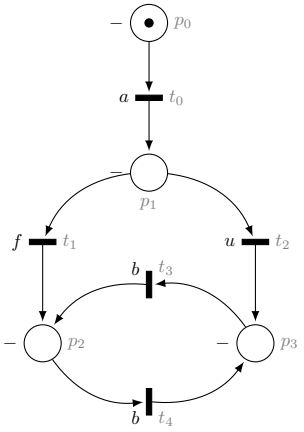

Fig. 4. Le système $\mathcal{N}$: f est un événement de faute, seuls a et b sont observables

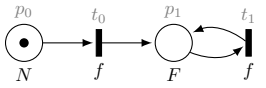

Fig. 5. Le motif de supervision Ω : occurrence d'un événement de faute f

La première étape consiste à faire le produit synchronisé des deux RdPÉ sur l'ensemble d'étiquettes $\Sigma_s = \Sigma \cap \Sigma_\Omega = \{f\}$, ce qui conduit au RdPÉ $\mathcal{N}_\Omega$ de la figure 6.

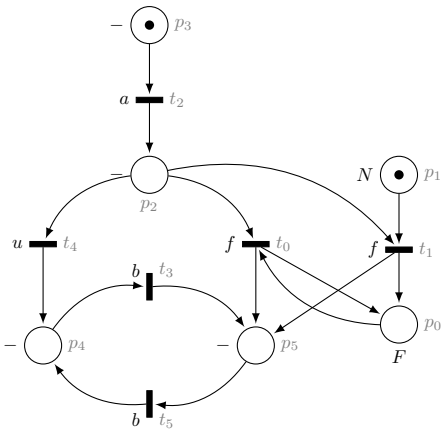

Fig. 6. Le produit $\mathcal{N}_\Omega = \mathcal{N} \|_{\Sigma_s} \Omega$

Ensuite, le produit de $\mathcal{N}_\Omega$ avec lui-même — sur l'ensemble des observables $\Sigma_o = \{a, b\}$ — est effectué pour obtenir le produit jumelé, le résultat Γ est représenté par la figure 7.

Enfin, la figure 8 représente le dépliage — du RdPÉ Γ — retourné par la fonction «dépliage» de l'algorithme 1. À noter que les éléments entourés par un cadre ne font pas vraiment partie du dépliage, ils ont été inclus dans la figure pour mieux visualiser les transitions de coupe lors de l'étape

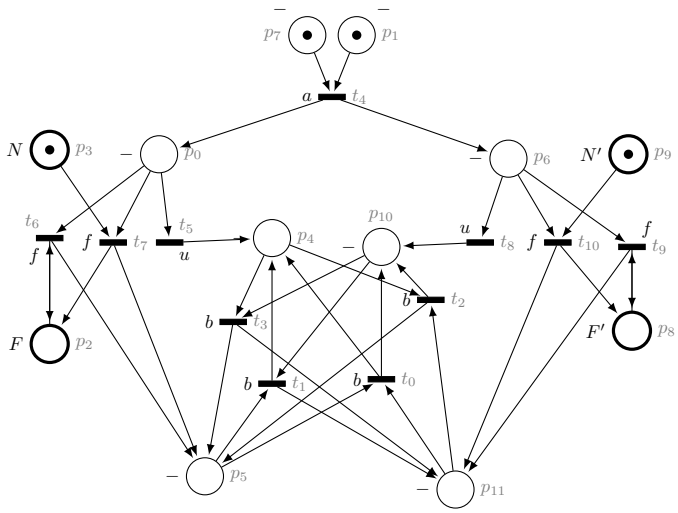

Fig. 7. Le produit jumelé $\Gamma = \mathcal{N}_\Omega \|_{\{a,b\}} \mathcal{N}'_\Omega$

d'analyse. Les étiquettes des places (resp. des transitions) apparaissent en noir dans le dépliage et correspondent aux noms des places (resp. des transitions) dans le produit jumelé Γ.

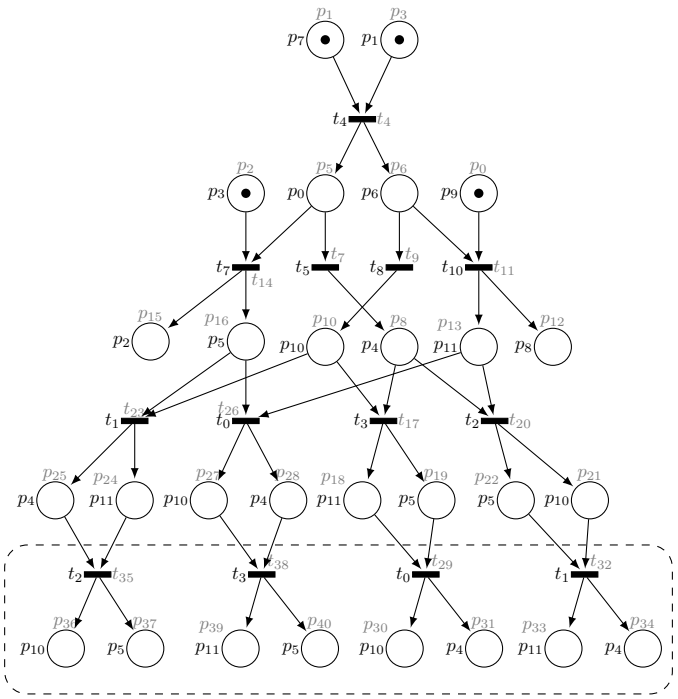

Fig. 8. Le dépliage du RdPÉ Γ

La fonction «dépliage» retourne — en plus du dépliage $\mathcal{D}$ — la table H qui associe à chaque transitions de coupe t — c.-à-d. à chaque transition incluse dans le cadre — le marquage résultant de son franchissement $H[t]$. Le tableau I liste les transitions de coupe de H et les marquages associés. Par exemple, le tir de la transition de coupe t_{32} depuis m_0 produit le marquage des places p_{34}, p_{33}, p_2 et p_{12}. La séquence de transitions conduisant au tir de t_{32} est indiquée en gras sur la figure 8.

Il ne reste ensuite qu'à effacer de H les éléments dont

TABLE I
LES TRANSITIONS DE H AVEC LES MARQUAGES ASSOCIÉS À LEUR FRANCHISSEMENT

transitions de coupe	marquage résultant
t_{29}	$\{p_{31}, p_{30}, p_2, p_0\}$
t_{32}	$\{p_{34}, p_{33}, p_2, p_{12}\}$
t_{35}	$\{p_{37}, p_{36}, p_{15}, p_0\}$
t_{38}	$\{p_{40}, p_{39}, p_{15}, p_{12}\}$

le marquage associé n'est pas ambigu, c.-à-d. les éléments $H[t_{29}]$ et $H[t_{38}]$. En effet, les éléments de $H[t_{29}]$ c.-à-d. les places $\{p_{30}, p_{31}, p_2, p_0\}$ du dépliage correspondent aux places $\{p_{10}, p_4, p_3, p_9\}$ du produit jumelé (voir figures 7 et 8), qui traduisent un état normal (étiquette N pour p_3 et N' pour p_9). De même, $H[t_{38}]$ traduit un état fautif (étiquette F pour p_2 et F' pour p_8).

Au final, H n'est pas vide — il reste t_{32} et t_{35} qui correspondent à des marquages ambigus, puisque incluant respectivement les places étiquetées N, F' et F, N' dans le produit jumelé (figure 7). Ceci correspond à l'existence — dans le système — de deux traces, l'une normale et l'autre fautive et qui ont le même comportement observable. Le système n'est donc pas diagnosticable vis-à-vis du motif de supervision choisi (voir section IV).

Ce résultat peut se voir facilement sur le système $\mathcal{N}$, il existe bien deux évolutions possibles, $s_1 = afbbb\ldots$ et $s_2 = aubbb\ldots$, l'une correspondant à un comportement de faute, l'autre à un comportement normal et dont le comportement observable $s_o = abbb\ldots$ est similaire.

Ce résultat peut aussi se déduire de manière systématique à partir de l'analyse des transitions de coupe restant dans la structure H et des traces conduisant à leur sensibilisation.

VII. DISCUSSION ET PERSPECTIVES

Dans cet article, nous proposons d'analyser la Ω-*diagnosticabilité* d'un motif de supervision en adaptant le concept de motif et de produit jumelé aux réseaux de Petri et en bénéficiant de la technique de dépliage. L'Ω-*diagnosticabilité* est un concept introduit dans [3]. Il démontre que la Ω-diagnosticabilité englobe la diagnosticabilité classique défini dans [1] (voir section VI, pour l'illustration du choix du motif vérifiant cela). [13], quant à lui, utilise les notions de K-diagnosticabilité (diagnosticabilité de K occurrence d'un événement) et $[1, K]$-diagnosticabilité (diagnosticabilité de J fois l'occurrence d'un événement avec $1 \leq J \leq K$). Ces notions sont un cas particulier de la Ω-diagnosticabilité, il suffit, pour vérifier la première, de construire un motif de supervision modélisant l'occurrence multiple (K fois) du même événement (voir sous-section III-B), et de construire plusieurs motifs pour vérifier la $[1, K]$-diagnosticabilité. Enfin, [13] propose aussi la notion de $[1, \infty]$-diagnosticabilité et l'algorithme qui permet de la vérifier.

La $[1, \infty]$-diagnosticabilité ne peut pas être vérifiée directement par la méthode de [3] puisqu'il faudrait construire une infinité de motifs. La méthode que nous proposons n'est pour la même raison pas applicable. Il serait donc intéressant de considérer une adaptation de l'approche pour traiter la $[1, \infty]$-diagnosticabilité.

Par ailleurs, dans l'optique de proposer une méthode d'analyse générique de la diagnosticabilité par dépliage de réseaux de Petri et fondée sur des motifs de supervision, nous envisageons d'étendre nos travaux aux motifs temporels et donc aux réseaux de Petri temporels. Cela nous permettra de prendre en compte des motifs de faute impliquant par exemple des délais ou l'absence d'événements particuliers pendant une certaine durée.

RÉFÉRENCES

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

[2] Shengbing Jiang and Ratnesh Kumar. Failure diagnosis of discrete-event systems with linear-time temporal logic specifications. *Transactions on Automatic Control*, 49(6) :934–945, 6 2004.

[3] Thierry Jéron, Hervé Marchand, Sophie Pinchinat, and Marie-Odile Cordier. Supervision Patterns in Discrete Event Systems Diagnosis. In *Workshop on Discrete Event Systems, WODES'06*, pages 262–268, Ann-Arbor, États-Unis, July 2006. IEEE Computer society.

[4] Albert Benveniste, Éric Fabre, Stefan Haar, and Claude Jard. Diagnosis of asynchronous discrete-event systems : a net unfolding approach. *Transactions on Automatic Control*, 48(5) :714–727, 5 2003.

[5] Francesco Basile, Pasquale Chiacchio, and Gianmaria De Tommasi. An efficient approach for online diagnosis of discrete event systems. *Transactions on Automatic Control*, 54(4) :748–759, 4 2009.

[6] Mariagrazia Dotoli, Maria Pia Fianti, Agostino Marcello Mangini, and Walter Ukovich. On-line fault detection in discrete event systems by petri nets and integer linear programming. *Automatica*, 45(11) :2665–2672, 11 2009.

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

[8] Javier Esparza, Stefan Römer, and Walter Vogler. An Improvement of McMillan's Unfolding Algorithm. *Formal Methods in System Design*, 20(3) :285–310, 2002.

[9] Maria Paola Cabasino, Alessandro Giua, Stéphane Lafortune, and Carla Seatzu. A new approach for diagnosability analysis of petri nets using verifier nets. *Transactions on Automatic Control*, 57(12) :3104–3117, 5 2012.

[10] Bernard Berthomieu, Florent Peres, and François Vernadat. Bridging the gap between timed automata and bounded time petri nets. In *FORMATS*, pages 82–97, 2006.

[11] Tae-Sic Yoo and Stéphane Lafortune. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Trans. Automat. Contr.*, 47(9) :1491–1495, 2002.

[12] Shengbing Jiang, Zhongdong Huang, Vigyan Chandra, and Ratnesh Kumar. A polynomial algorithm for testing diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 2000.

[13] Shengbing Jiang, Ratnesh Kumar, and Humberto E. Garcia. Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Transactions on Robotics*, 19(2) :310–323, 2003.