
Diagnostic Décentralisé des Systèmes à Événements Discrets

Stéphane Lafortune* — Yin Wang* — Tae-Sic Yoo**

* *Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, Michigan 48109-2122, USA.
www.eecs.umich.edu/umdcs - [stephane,yinw]@umich.edu*

** *Idaho National Laboratory, Idaho Falls, Idaho 83403, USA*

RÉSUMÉ. Cet article traite du problème de la détection et de l'identification d'événements inobservables dans le comportement des systèmes à événements discrets. Nous nous intéressons aux méthodes de diagnostic dites décentralisées où plusieurs sites observent le comportement du système, chaque site ayant son propre ensemble d'événements observables. Les sites ne peuvent pas communiquer entre eux durant le diagnostic mais leurs décisions de diagnostic respectives sont fusionnées en temps réel par une fonction booléenne sans mémoire. Nous présentons plusieurs nouveaux résultats dans le cadre de telles architectures, en mettant l'accent sur l'emploi de décisions conditionnelles par les sites locaux.

ABSTRACT. We consider the detection and identification of unobservable events in the behavior of discrete event systems. We study decentralized diagnosis architectures where several sites observe the system behavior, each site having its own set of observable events. These sites do not communicate among each other but they send their local diagnostic decisions to a coordinating site that implements a simple memoryless Boolean fusion rule. Several new results are presented in the context of such architectures, with special focus on the use of conditional decisions by the local sites.

MOTS-CLÉS : diagnostic décentralisé, diagnostiquabilité, codiagnostiquabilité, diagnostiquabilité conditionnelle.

KEYWORDS: decentralized diagnosis, diagnosability, codiagnosability, conditional diagnosability.

1. Introduction

Le diagnostic des systèmes à événements discrets est le problème de détecter et d'identifier certains événements inobservables qui ont lieu durant le fonctionnement du système en se basant sur le modèle du système et sur les séquences d'événements observables obtenues des capteurs reliés au système. Les événements inobservables à détecter font parti du modèle du système et représentent par exemple des fautes ou d'autres anomalies dans le comportement du système. Ce domaine de recherche a reçu beaucoup d'attention durant le dernière décennie. Parmi le grand nombre de travaux qui ont traité de ce sujet, nous mentionnons les articles suivants qui sont pertinents au résultats qui suivent : (Sampath *et al.*, 1995, Sampath *et al.*, 1996, Sengupta, 1998, Debouk *et al.*, 2000, Fabre *et al.*, 2000, Lafortune *et al.*, 2001, Boel *et al.*, 2002, Sengupta *et al.*, 2002, Su *et al.*, 2002, Yoo *et al.*, 2002b, Benveniste *et al.*, 2003, Lamperti *et al.*, 2003, Boel *et al.*, 2004, Qiu *et al.*, 2004, Su *et al.*, 2004, Genc *et al.*, 2005).

Les méthodes de diagnostic dites décentralisées deviennent nécessaires lorsqu'on considère des systèmes à événements discrets où l'information est décentralisée. Ceci est le cas lorsque le système a une architecture répartie où plusieurs sites observent son comportement. Chaque site reçoit les observations d'un sous-ensemble de l'ensemble des capteurs et doit faire son propre diagnostic en se basant sur ses observations et sur le modèle de l'ensemble du système. Les sites ne peuvent pas échanger entre eux des messages durant l'évolution du système. Par contre, les résultats du diagnostic local à chaque site sont communiqués à un coordonnateur où les diagnostics locaux sont fusionnés. Nous nous intéressons au cas où le coordonnateur a une structure qui est la plus simple possible, c'est-à-dire qu'il consiste d'une simple fonction booléenne sans mémoire. Cette architecture est naturelle dans le cas de systèmes répartis tels que les réseaux de communication ; elle fut étudiée dans de nombreux articles incluant par exemple (Sengupta, 1998, Debouk *et al.*, 2000). Le Protocole 3 présenté dans (Debouk *et al.*, 2000) est un cas limite de cette architecture où le rôle du coordonnateur est réduit au minimum : celui-ci ne fait que transmettre directement les messages reçus des sites locaux sans aucune fusion explicite de ces messages.

Notre objectif est d'élargir la classe de systèmes qui sont diagnostiquables par le Protocole 3 de (Debouk *et al.*, 2000) en utilisant des fonctions booléennes sans mémoire au site du coordonnateur. Pour atteindre cet objectif, il faut choisir un ensemble de décisions locales aux différents sites du système qui vont de pair avec la fonction de fusion de ces décisions employée par le coordonnateur. À cet égard, nous présentons dans les sections qui suivent de nouvelles stratégies de décisions dites « inconditionnelles » et « conditionnelles » qui peuvent être employées par les sites locaux pour permettre de diagnostiquer des systèmes qui ne sont pas diagnostiquables par le Protocole 3 de (Debouk *et al.*, 2000). Notre approche requiert l'emploi de décisions positives *et* négatives à chaque site concernant la présence d'une faute (c'est-à-dire d'un événement inobservable qu'il faut détecter) dans le comportement du système ;

nous avons donc les deux décisions locales FAUTE (F) et ABSENCE DE FAUTE (NF)¹. Par la suite, nous ajoutons les décisions conditionnelles F SI AUCUN SITE NE DIT NF et NF SI AUCUN SITE NE DIT F. Nous démontrons que l'emploi de telles décisions conditionnelles, avec leur fonction de fusion au site du coordonnateur, est un outil puissant pour diagnostiquer les systèmes dans le cadre d'une architecture décentralisée sans aucune communication en temps réel entre les sites (autre que les décisions mentionnées ci-haut).

Notre approche est en partie inspirée par les travaux récents dans (Yoo *et al.*, 2002a, Yoo *et al.*, 2004) sur le *contrôle* décentralisé des systèmes avec des décisions inconditionnelles et conditionnelles quant à la commande des événements contrôlables (« enable if nobody disables » et « disable if nobody enables » dans la terminologie de (Yoo *et al.*, 2004)). Il s'avère que cette approche est non seulement avantageuse pour le contrôle des systèmes à événements discrets mais également pour leur diagnostic. Nous notons que notre approche diffère de celle qui a inspiré les Protocoles 1 et 2 dans (Debouk *et al.*, 2000) où la fonction de fusion employée par le coordonnateur est basée sur l'intersection des états des diagnostiqueurs locaux (avec mémoire dans le cas du Protocole 1) qui sont communiqués au site du coordonnateur. Dans notre approche, les sites locaux ne communiquent pas les états de leurs diagnostiqueurs respectifs, mais seulement leurs décisions quant à la présence ou l'absence de l'événement de faute : F, NF, F SI AUCUN NF, NF SI AUCUN F.

Cet article est structuré comme suit. Le problème de diagnostiquer l'absence de fautes est considéré dans la Section 3. Les sections qui suivent traitent du diagnostic décentralisé : revue (Section 4), cas des décisions inconditionnelles (Section 5) et cas des décisions conditionnelles (Section 6). Nous présentons d'abord les notations et résultats préliminaires nécessaires pour le reste de l'article dans la Section 2.

2. Préliminaires

Étant donné le nombre limité de pages, notre revue des travaux précédents et des définitions de base est réduite au minimum. Le lecteur est invité à consulter (Lafortune *et al.*, 2001) pour une revue de la littérature sur le diagnostic des systèmes à événements discrets et (Cassandras *et al.*, 1999) pour certains concepts de base.

Considérons un système à événements discrets modélisé par un langage régulier $L \subseteq E^*$ où E est l'ensemble des événements du système. Le langage L est fermé au sens des préfixes ($L = \overline{L}$ dans la notation usuelle) et représenté par la notation des expressions régulières (dans les exemples qui suivent) ou, de façon équivalente, par un automate à états finis $G = (X, E, f, q_0)$, où X est l'espace d'états, f est la fonction partielle de transition $f : X \times E \rightarrow X$, et x_0 est l'état initial. L est le langage généré par G . Le langage L modélise non seulement le comportement normal du système, mais aussi son comportement après que certains événements dits de faute se produisent.

1. Nous utilisons le même acronyme que l'expression anglaise « No Fault ».

L'ensemble E des événements est partitionné en événements observables E_o et inobservables E_{uo} : $E = E_o \cup E_{uo}$. Les événements inobservables sont ceux qui ne sont pas directement mesurés par les capteurs reliés au système. L'ensemble des événements qu'il faut détecter et identifier, souvent appelé l'ensemble des événements de faute et dénoté par E_F , est un sous-ensemble de E_{uo} . Nous considérons d'abord le cas où $E_F = \{e_f\}$ est un singleton car il suffit pour présenter plusieurs des concepts et résultats de cet article. Nous discuterons plus tard le cas où il y a plusieurs événements de faute, $E_F = \{e_{f1}, \dots, e_{fM}\}$. Une *trace* $s \in L$ est dite *fautive* si elle est de la forme $s = ue_fv$ où u et v sont dans E^* . Donc s est fautive si elle contient l'événement de faute e_f .

Considérons l'opération habituelle de projection P de E^* à E_o^* qui *efface* les événements inobservables dans une trace ; voir (Cassandras *et al.*, 1999) pour la définition complète. Nous avons que $P^{-1}(s) := \{t \in E^* : P(t) = s\}$. Nous employons la notation $\mathcal{E}(s) = P^{-1}P(s) \cap L$ pour dénoter l'ensemble des traces du système qui ont la même projection (c'est-à-dire la même sous-séquence d'événements observables) que la trace s . $\mathcal{E}(s)$ correspond donc à l'estimation du comportement du système après l'observation de la trace $P(s)$. Par conséquent, $t \in \mathcal{E}(s)$ ssi $t \in L$ et $P(t) = P(s)$.

Afin de simplifier la présentation qui suit, nous faisons les deux hypothèses suivantes qui sont standard dans la littérature : **(A1)** L est vivant et **(A2)** Chaque cycle dans le graphe de G contient au moins un événement observable. Il est facile d'éliminer ces hypothèses au prix d'une notation alourdie.

Notre point de départ est la définition de la propriété de *diagnostiquabilité* pour les systèmes à événements discrets introduite dans (Sampath *et al.*, 1995, Sampath *et al.*, 1996). Sous les hypothèses précédentes et dans le cas où $E_F = \{e_f\}$, nous avons la définition suivante.

Définition 1 *Le langage L est diagnostiquable, dénoté par F-DIAG, si la condition suivante est vérifiée :*

$$(\exists k \in \mathbb{N}) (\forall s \in L : s \text{ est fautive}) (\forall st \in L : |t| \geq k) (\forall u \in \mathcal{E}(st)) u \text{ est fautive.}$$

Cette définition s'explique comme suit. Si s est une trace fautive et t une continuation de s qui est suffisamment longue, alors toute trace dans L qui a la même projection que st doit contenir e_f . La notion de F-DIAG veut donc dire qu'il est toujours possible de diagnostiquer e_f dans un délai borné après que cet événement ait lieu.

3. Diagnostiquer l'Absence de Fautes

La notion de F-DIAG revue dans la section précédente caractérise le fait que toute exécution de l'événement de faute e_f par le système est éventuellement diagnostiquable d'après les sous-séquences observables des traces qu'il génère en se basant sur le modèle du système. Il s'avère intéressant de considérer le problème connexe où nous devons diagnostiquer l'absence de l'événement e_f dans le comportement du

système, toujours d'après les sous-séquences d'événements observables. Nous appelons cette notion celle de la diagnostiquabilité de l'absence de fautes, dénotée par NF-DIAG ci-après. Nos travaux ont conduit à plusieurs variantes de la notion de NF-DIAG qui ont chacune des propriétés intéressantes (Wang *et al.*, 2005). Nous présentons ici seulement une des définitions de NF-DIAG qui sera suffisante pour les besoins de cet article ; il s'agit de NF-DIAG-3 dans la terminologie de (Wang *et al.*, 2005).

Définition 2 *Le langage L a la propriété NF-DIAG-3 si la condition suivante est vérifiée :*

$$(\exists k \in \mathbb{N}) (\forall s \in L : s \text{ est non-fautive}) (\forall st \in L : |t| \geq k \text{ et } st \text{ est non-fautive}) \\ (\forall uv \in \mathcal{E}(st) : P(u) = P(s)) \text{ } u \text{ est non-fautive.}$$

L'interprétation de cette définition est la suivante. Soit s une trace non-fautive exécutée par le système et soit t un suffixe non-fautif de s qui est suffisamment long. Alors quelle que soit la trace que le système ait exécutée parmi toutes celles qui ont la projection $P(st)$, celle-ci est non-fautive au moins jusqu'à son préfixe $P(s)$. Autrement dit, si le système fonctionne sans jamais exécuter l'événement de faute e_f , alors on peut toujours diagnostiquer que le comportement du système était non-fautif à un certain moment dans le passé.

Exemple 1 Soit le système modélisé par le langage $\overline{a^*fab^*}$ où le seul événement inobservable est l'événement de faute f . Ce système est NF-DIAG3. Soit $k = 1$ et considérons la trace non-fautive a^n (qui est la seule sans faute). Alors $\mathcal{E}(a^n) = \{a^n, a^n f, a^{n-1} f a\}$, $u \in \{a^n, a^n f, a^{n-1} f a\}$, $|u| \leq |st| - k = n - 1$, impliquent que $u = a^{n-1}$ et est donc non-fautif. ■

La propriété de NF-DIAG-3 signifie également que si une trace fautive s et une trace non-fautive u ont la même projection, alors après au plus k événements après l'exécution de la faute, les continuations de s et de u doivent différer par des événements observables. Autrement dit, toute faute doit devenir diagnostiquable après un suffixe borné. Nous avons donc le théorème suivant.

Théorème 1 *L est F-DIAG ssi L est NF-DIAG-3.*

Preuve: \neg NF-DIAG-3 \Rightarrow \neg F-DIAG. Toute violation de NF-DIAG-3 implique l'existence d'une trace $uv \in \mathcal{E}(st)$ telle que u est fautive et $P(u) = P(s)$. Alors $P(v) = P(t)$ et $|t| \geq k$, où l'entier k peut être choisi arbitrairement grand. Sous l'hypothèse (A2), v et t sont arbitrairement longs. Par conséquent, u est fautive et possède un suffixe arbitrairement long v tel que $P(uv) = P(st)$ et st est non-fautive. Donc L n'est pas F-DIAG.

\neg F-DIAG \Rightarrow \neg NF-DIAG-3. Cette preuve est similaire et omise. ■

Grace au Théorème 1, les méthodes développées pour tester F-DIAG peuvent être employées pour tester NF-DIAG-3. Ces méthodes sont basées soit sur la construction

d'un automate déterministe appelé le *diagnostiqueur* (voir (Sampath *et al.*, 1995)), soit sur la construction d'un automate non-déterministe appelé le *vérificateur*² (Yoo *et al.*, 2002b). (On remarque que la méthode du vérificateur est elle-même basée sur la méthodologie de (Rudie *et al.*, 1995) pour la notion de coobservabilité.) Le test basé sur le vérificateur est polynomial en $|X|$, l'espace d'états de G , ce qui n'est pas le cas pour le test basé sur le diagnostiqueur. Par contre, le diagnostiqueur sert aussi à l'implémentation en temps réel du diagnostic des fautes, tel que décrit dans (Sampath *et al.*, 1995). On peut démontrer qu'on peut aussi se servir du même diagnostiqueur pour implémenter le diagnostic de l'absence de fautes pour les systèmes qui sont NF-DIAG-3, quoique les détails sont plutôt compliqués ; voir (Wang *et al.*, 2005).

4. Diagnostic Décentralisé

Considérons l'architecture de diagnostic dite décentralisée de la Fig. 1 où les n sites locaux associés au système G observent chacun un sous-ensemble de l'ensemble E_o des événements observables de G . L'ensemble $E_{o,i}$ est l'ensemble des événements observables au site i , $i = 1, \dots, n$. Les blocs P_i dans la figure représentent les opérations de projections de E^* à $E_{o,i}^*$ qui effacent les événements inobservables au site i . Le bloc de fusion dans la figure consiste en une fonction booléenne sans mémoire qui fusionne les décisions de diagnostic locales qui lui sont transmises. Tel que mentionné dans la Section 1, nous excluons les fonctions de fusion plus complexes, possiblement avec mémoire, comme celles des Protocoles 1 et 2 de (Debouk *et al.*, 2000). Nous excluons aussi une communication directe entre les fonctions locales, un cas étudié dans (Boel *et al.*, 2002). Notre but est d'étudier les architectures de décisions décentralisées (sans communication) avec les blocs de fusion les plus simples possibles, de telle sorte qu'on puisse mieux comprendre comment optimiser les processus d'inférence locaux à chaque site. C'est dans ce contexte qu'il faut interpréter les méthodes avec décisions conditionnelles qui seront présentées dans la Section 6. Ces méthodes sont elles-mêmes basées sur les décisions dites inconditionnelles traitées dans la Section 5.

La fonction \mathcal{E} de la Section 2 est généralisée au cas décentralisé de façon naturelle : $\mathcal{E}_i(s) = P_i^{-1}P_i(s) \cap L$ où nous avons que $P_i^{-1}(s) := \{t \in E^* : P_i(t) = s\}$.

La définition qui suit porte sur un langage L qui satisfait les hypothèses (A1) et (A2) de la Section 2 et qui contient un seul événement de faute e_f . Cette définition présume que L est généré par le système G et que le diagnostic est effectué dans le contexte de l'architecture de la Fig. 1.

Définition 3 *Le langage L est F-codiagnostiquable, dénoté par F-CODIAG, si la condition suivante est vérifiée :*

$(\exists k \in \mathbb{N}) (\forall s \in L : s \text{ est fautive}) (\forall st \in L : |t| \geq k) (\exists i \in \{1, \dots, n\}) (\forall u \in \mathcal{E}_i(st)) u \text{ est fautive.}$

2. « Verifier » dans la littérature en anglais.

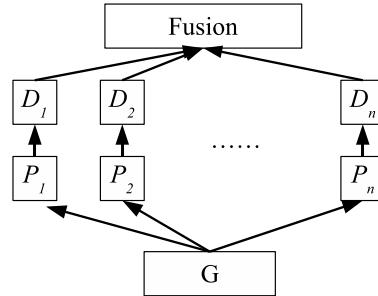


Figure 1. Architecture décentralisée

L'interprétation de cette définition est comme suit. Soit une trace s fautive et un suffixe t de cette trace qui soit suffisamment long, avec $st \in L$. Alors il doit exister au moins un site i , $i \in \{1, \dots, n\}$, tel que toute trace de L ayant la même projection que st au site i soit aussi fautive. Autrement dit, lorsque L est F-CODIAG, chaque exécution de l'événement de faute e_f sera diagnostiquée par au moins un des sites locaux. Cette définition est identique à la notion de « diagnostiquabilité dans le cadre du Protocole 3 » de (Debouk *et al.*, 2000). D'ailleurs, cette définition est aussi reprise dans (Qiu *et al.*, 2004) sous le nom de « co-diagnostiquabilité ». Nous adoptons dans cet article le nom *F-codiagnostiquabilité* de façon à obtenir une terminologie consistante dans cet article et aussi pour faciliter les comparaisons entre nos travaux et ceux dans (Yoo *et al.*, 2002a, Yoo *et al.*, 2004) qui portent sur la propriété de *coobservabilité* et de son rôle en contrôle décentralisé.

La notion de F-CODIAG est adaptée au cas où la seule décision locale utilisée pour le diagnostic à chaque site est la décision FAUTE (F). Par conséquent, le système est diagnostiquable si au moins un des sites locaux déclare F. La fonction de fusion correspondant à la définition de F-CODIAG est donc triviale. Le lecteur familier avec les diagnostiqueurs de (Sampath *et al.*, 1995) et leur emploi dans (Debouk *et al.*, 2000) dans les architectures décentralisées se rendra compte que dans le contexte de F-CODIAG, il suffit pour chaque site i d'utiliser le diagnostiqueur pour L correspondant à l'ensemble d'événements observables $E_{o,i}$; cf. le Protocole 3 de (Debouk *et al.*, 2000).

5. Diagnostic Décentralisé : Cas de l'Absence de Fautes

5.1. Notions de Codiagnosabilité

Nous nous intéressons dans cette section à la détection de *l'absence* de fautes dans le cadre de l'architecture de la Fig. 1 et nous introduisons la notion de *NF-codiagnosabilité*. Considérons l'exemple qui suit.

Exemple 2 Soit le système modélisé par la langage $\overline{(f + a + b)c^*}$ où $E_o = \{a, b, c\}$ and $E_F = E_{uo} = \{f\}$. Les capteurs du système sont répartis sur deux sites, $n = 2$, avec $E_{o,1} = \{a, c\}$ and $E_{o,2} = \{b, c\}$.

Ce système n'est pas F-CODIAG parce que la trace fautive fc^m , qui est de longueur arbitraire, a la même projection que la trace non-fautive bc^m au site 1 et a la même projection que la trace non-fautive ac^m au site 2. Par contre, ces dernières traces non-fautives peuvent être diagnostiquées par un site ou l'autre. Par exemple, si le site 1 observe l'événement a , ceci indique que l'événement f n'a pas eu lieu. De même pour une observation de l'événement b au site 2. ■

En s'inspirant de l'exemple 2 ainsi que de la notion d'architecture *disjonctive* pour le contrôle décentralisé de systèmes à événements discrets introduite dans (Yoo *et al.*, 2002a), nous proposons le nouveau concept de NF-codiagnosabilité, dénoté par NF-CODIAG, qui permet aux sites locaux d'émettre la décision de diagnostic ABSENCE DE FAUTE (NF). Ceci nous conduit à la définition suivante.

Définition 4 *Le langage L est NF-CODIAG si la condition suivante est vérifiée :*
 $(\exists k \in \mathbb{N}) (\forall s \in L : s \text{ est non-fautive}) (\forall st \in L : |t| \geq k \text{ et } st \text{ est non-fautive})$
 $(\exists i \in \{1, 2, \dots, n\}) (\forall uv \in \mathcal{E}_i(st) : P_i(u) = P_i(s)) \text{ } u \text{ est non-fautive.}$

Cette définition a rapport à la détection *décentralisée de l'absence* de fautes. Soit s une trace non-fautive et soit t un suffixe non-fautif de s de longueur arbitraire ; alors il doit exister un site i pour lequel toute trace qui a la même projection P_i que st est aussi non-fautive au point où elle a la même projection que s . On peut constater que la notion de NF-CODIAG est l'extension au cas décentralisé de la notion NF-DIAG-3 introduite plus tôt dans la Section 3.

Il est facile de vérifier que le système de l'exemple 2 ci-haut est NF-CODIAG. Les traces non-fautives avec des suffixes de longueur arbitraire sont ac^m et bc^m , et ces traces seront diagnostiquées correctement comme étant non-fautives par les sites 1 et 2, respectivement.

Dans le cas général où il y a un *ensemble* d'événements de faute à diagnostiquer, $E_F = \{e_{f1}, \dots, e_{fM}\} \subseteq E_{uo}$, nous pouvons généraliser les définitions des propriétés F-CODIAG et NF-CODIAG. Nous utiliserons la terminologie *Fj-fautive* pour indiquer la présence de l'événement e_{fj} dans une trace.

Définition 5 *Le langage L est F-CODIAG par rapport à $\{e_{f1}, \dots, e_{fK}\}$ si la condition suivante est vérifiée :*
 $(\forall j \in \{1, \dots, K\}) (\exists k_j \in \mathbb{N}) (\forall s \in L : s \text{ est } Fj\text{-fautive}) (\forall st \in L : |t| \geq k_j)$
 $(\exists i \in \{1, 2, \dots, n\}) (\forall u \in \mathcal{E}_i(st)) \text{ } u \text{ est } Fj\text{-fautive.}$

Définition 6 *Le langage L est NF-CODIAG par rapport à $\{e_{f1}, \dots, e_{fK}\}$ si la condition suivante est vérifiée :*

$(\forall j \in \{1, \dots, K\}) (\exists k_j \in \mathbb{N}) (\forall s \in L : s \text{ n'est pas } Fj\text{-fautive}) (\forall st \in L : |t| \geq k_j \text{ et } st \text{ n'est pas } Fj\text{-fautive}) (\exists i \in \{1, 2, \dots, n\}) (\forall uv \in \mathcal{E}_i(st) : P_i(u) = P_i(s)) \text{ u n'est pas } Fj\text{-fautive}.$

Si un langage est F-CODIAG par rapport à l'ensemble de toutes les fautes E_F , alors nous dirons simplement que L est F-CODIAG. De même pour NF-CODIAG. Dans le but de capter la situation où chaque événement de E_F est soit F-CODIAG, soit NF-CODIAG (ou les deux), nous partitionnons l'ensemble des fautes E_F en $E_F = E_{F,F} \cup E_{F,NF}$, où $E_{F,F}$ est l'ensemble des fautes dont la présence est diagnostiquable et $E_{F,NF}$ est l'ensemble des fautes dont l'absence est diagnostiquable. Si une telle partition existe, alors nous avons la définition suivante.

Définition 7 L est codiagnostiquable, dénoté par CO-DIAG, par rapport à $E_{F,F}$ et $E_{F,NF}$ si :

1. L est F-CODIAG par rapport à $E_{F,F}$; et
2. L est NF-CODIAG par rapport à $E_{F,NF}$.

5.2. Propriétés des Notions de Codiagnostiquabilité

Théorème 2 F-CODIAG et NF-CODIAG sont des propriétés incomparables.

L'exemple 2 ci-haut démontre un cas où F-CODIAG n'est pas satisfaite mais NF-CODIAG l'est. L'exemple 3 qui suit démontre la situation inverse.

Exemple 3 Considérons le système modélisé par le langage $\overline{c^*f(a+b)c^*}$ où $E_o = \{a, b, c\}$ et $E_{uo} = E_F = \{f\}$. Il y a deux sites : $E_{o,1} = \{a, c\}$ et $E_{o,2} = \{b, c\}$. Ce système est F-CODIAG puisque les traces fautives c^mfac^r et c^mfbcr seront diagnostiquées par les sites 1 et 2, respectivement. Par contre, ce système n'est pas NF-CODIAG parce que la trace non-fautive de longueur arbitraire c^m ne sera pas diagnostiquée par le site 1 qui la confond avec la trace fautive $fbcm$, ni par le site 2 qui la confond avec la trace fautive fac^m . ■

Théorème 3 F-CODIAG ou NF-CODIAG implique codiagnostiquable. Par contre, l'inverse n'est pas vrai en général.

La première partie de ce théorème est une conséquence des définitions respectives de ces trois propriétés. La deuxième partie est démontrée par l'exemple 4 qui suit.

Exemple 4 Considérons le système modélisé par le langage

$$\overline{c_1^*[f_1(a_1 + b_1)c_1^* + (f_2 + a_2 + b_2)c_2^*]}$$

où $E_o = \{a_1, a_2, b_1, b_2, c_1, c_2\}$ et $E_{uo} = E_F = \{f_1, f_2\}$. E_F est partitionné en $E_{F1} = \{f_1\}$ et $E_{F2} = \{f_2\}$. Il y a deux sites : $E_{o,1} = \{a_1, a_2, c_1, c_2\}$ et

$E_{o,2} = \{b_1, b_2, c_1, c_2\}$. D'après les exemples 2 et 3, nous savons que ce système est codiagnostiquable avec la partition suivante : $E_{F,F} = \{f_1\}$ et $E_{F,NF} = \{f_2\}$. (Autrement dit, f_1 est F-CODIAG et f_2 est NF-CODIAG.) Par contre, f_2 n'est pas F-CODIAG et f_1 n'est pas NF-CODIAG. ■

Théorème 4 *Codiagnostiquabilité par rapport à $E_{F,F}$ et $E_{F,NF}$ implique diagnostiquabilité (au sens centralisé, ou F-DIAG) pour tous les événements dans $E_{F,F} \cup E_{F,NF}$. Par contre, l'inverse n'est pas vrai en général.*

Ce théorème est démontré comme suit. D'une part, tout événement de faute qui est F-CODIAG est nécessairement F-DIAG. De même, tout événement qui est NF-CODIAG est NF-DIAG-3. D'après le Théorème 1, NF-DIAG-3 est équivalente à F-DIAG. Donc la codiagnostiquabilité implique la diagnostiquabilité. L'exemple 5 qui suit prouve que l'implication inverse n'est pas vraie en général.

Exemple 5 Considérons le système modélisé par le langage $\overline{(fab + ba)c^*}$ où $E_o = \{a, b, c\}$ et $E_{uo} = E_F = \{f\}$. Il y a deux sites : $E_{o,1} = \{a, c\}$ et $E_{o,2} = \{b, c\}$. Ce système n'est pas codiagnostiquable puisque peu importe si f a lieu ou non, le site 1 observe toujours ac^m et le site 2 observe toujours bc^m . ■

5.3. Vérification des Propriétés de Codiagnostiquabilité

Il est possible de démontrer que toutes les notions de codiagnostiquabilité présentées ci-haut sont vérifiables avec une complexité de calcul polynomiale. Cela s'avère possible en généralisant et adaptant les méthodes respectives basées sur le vérificateur pour les notions de F-DIAG et de NF-DIAG-3. Nous référons le lecteur au rapport (Wang *et al.*, 2005) pour les détails de cette approche. Nous mentionnons deux remarques : (i) La grandeur du vérificateur (espace d'états, pire cas) est polynomiale par rapport au nombre d'états du système mais exponentielle par rapport au nombre de sites ; et (ii) Lorsque E_F contient plus d'une faute, il est suffisant de construire un vérificateur pour chaque événement de faute.

6. Diagnostic Décentralisé avec Décisions Conditionnelles

Les résultats de la section précédente sont basés sur une architecture décentralisée où les seules décisions locales permises sont FAUTE (F) et ABSENCE DE FAUTE (NF). Le bloc de fusion de ces décisions, tel qu'illustré dans la Fig. 1, est d'ailleurs trivial. Nous avons vu dans l'exemple 2 que dans ces conditions l'événement de faute était NF-CODIAG mais pas F-CODIAG. Il est possible de diagnostiquer la présence de la faute dans cet exemple si on utilise un ensemble plus riche de décisions locales. Dans ce but, nous introduisons maintenant des décisions locales dites *conditionnelles* de la forme : « Faute si aucun site ne dit Absence de Faute » (F SI AUCUN NF) et « Absence

de Faute si aucun site ne dit Faute » (NF SI AUCUN F). Nous nommons l'architecture décentralisée où de telles décisions sont utilisées *l'architecture conditionnelle*. Une telle architecture a été étudiée récemment dans le cadre du contrôle des systèmes à événements discrets dans (Yoo *et al.*, 2004).

Nous adoptons les règles de fusion des décisions locales pour le bloc de fusion, incluant décisions inconditionnelles et conditionnelles, qui sont énumérées au Tableau 1 qui suit. Tel qu'on peut le voir dans ce tableau, les décisions locales dans les cas 3 à 8 peuvent être interprétées comme des décisions F et NF mais avec une priorité moindre que les décisions inconditionnelles. Ces dernières annullent d'ailleurs les décisions conditionnelles comme on peut le voir dans les cas 4 et 7. En ce qui concerne les cas 10 et 11 où des conflits ont lieu, il s'agit là de situations à éviter lors du design des fonctions de diagnostic locales.

<i>Cas</i>	<i>Décision locale 1</i>	<i>Décision locale 2</i>	<i>Décision globale</i>
1	F	pas de décision	F
2	NF	pas de décision	NF
3	F si aucun NF	pas de décision	F
4	F si aucun NF	NF	NF
5	F si aucun NF	F	F
6	NF si aucun F	pas de décision	NF
7	NF si aucun F	F	F
8	NF si aucun F	NF	NF
9	pas de décision	pas de décision	pas de décision
10	F	NF	conflit
11	F si aucun NF	NF si aucun F	conflit

Tableau 1. *Décisions locales et globale dans l'architecture conditionnelle*

6.1. *Notions de Codiagnostiquabilité Conditionnelle*

De façon à établir un parallèle avec les résultats de la Section 5 (et aussi jusqu'à un certain point avec ceux de (Yoo *et al.*, 2004) dans le cadre du contrôle décentralisé), nous étudions d'abord deux cas spéciaux de l'architecture conditionnelle décrite au Tableau 1 : *F-codiagnostiquabilité conditionnelle* dans le cas spécial appelé architecture conditionnelle F (AC-F ci-après) et *NF-codiagnostiquabilité conditionnelle* dans le cas spécial appelé architecture conditionnelle NF (AC-NF ci-après).

Dans le cas AC-F, les sites locaux utilisent trois types de décisions : F, NF, et F SI AUCUN NF. Ces dernières correspondent aux cas 1, 2, 3, 4, 5, et 9 dans le Tableau 1. Notre première notion de codiagnostiquabilité conditionnelle est la suivante.

Définition 8 *Le langage L est conditionnellement F-codiagnostiquable, ou COND-F-CODIAG, si la condition suivante est vérifiée :*

$$(\exists k \in \mathbb{N}) (\forall s \in L : s \text{ est fautive}) (\forall st \in L : |t| \geq k) (\exists i \in \{1, \dots, n\}) (\forall uv \in \mathcal{E}_i(st) : P_i(u) = P_i(s) \text{ et } uv \text{ est non-fautive}) (\exists j \in \{1, \dots, n\}) (\forall xy \in \mathcal{E}_j(uv) : P_j(x) = P_j(u)) x \text{ est non-fautive}.$$

Cette définition a l'interprétation suivante. Pour chaque trace fautive st qui est suffisamment longue, il existe un site i pour lequel st est diagnostiquable comme étant fautive ou bien pour toutes les traces non-fautives de la forme uv qui ont la même projection que st au site i , le site i sait qu'il existe un site j pour lequel la trace uv sera diagnostiquée comme étant non-fautive jusqu'au moment où le préfixe u a eu lieu. Autrement dit, le site i « sait » que si le système a exécuté la trace uv et non la trace st , alors le site j va émettre la décision NF. Donc le site i choisit dans ce cas la décision F SI AUCUN NF.

Dans le cas dual de l'architecture AC-NF, les sites locaux utilisent trois types de décisions : F, NF, et NF SI AUCUN F. Ces dernières correspondent aux cas 1, 2, 6, 7, 8 et 9 dans le Tableau 1. Nous avons la définition suivante pour la notion de NF-codiagnostiquabilité conditionnelle.

Définition 9 *Le langage L est conditionnellement NF-codiagnostiquable, ou COND-NF-CODIAG, si la condition suivante est vérifiée :*

$$(\exists k \in \mathbb{N}) (\forall s \in L : s \text{ est non-fautive}) (\forall st \in L : |t| \geq k \text{ et } st \text{ est non-fautive}) (\exists i \in \{1, \dots, n\}) (\forall uv \in \mathcal{E}_i(st) : P_i(u) = P_i(s) \text{ et } u \text{ est fautive}) (\exists j \in \{1, \dots, n\}) (\forall w \in \mathcal{E}_j(uv)) w \text{ est fautive}.$$

L'interprétation de cette définition est similaire à celle de COND-F-CODIAG. Pour chaque trace non-fautive st qui est suffisamment longue, il existe un site i pour lequel st est diagnostiquable comme étant non-fautive ou bien pour toutes les traces dans $\mathcal{E}(st)$ de la forme uv où u est fautive et a la même projection que s au site i , le site i peut déterminer qu'il existe un site j pour lequel la trace u sera diagnostiquée comme étant fautive (puisque son suffixe v est suffisamment long). Autrement dit, le site i « sait » que si le système a exécuté la trace fautive u et non la trace non-fautive s , alors le site j va émettre la décision F. Donc le site i choisit dans ce cas la décision NF SI AUCUN F.

Les définitions de COND-F-CODIAG et COND-NF-CODIAG sont généralisées au cas de fautes multiples de la même façon que dans les Définitions 5 et 6 dans le cas de l'architecture sans décisions conditionnelles. Puisque ces extensions sont tout à fait similaires au cas inconditionnel, nous omettons de les présenter et passons directement au cas général de la propriété de codiagnostiquabilité conditionnelle. Soit $E_{F,F}$ l'ensemble des fautes dont la présence est diagnostiquable en utilisant possiblement des décisions conditionnelles et soit $E_{F,NF}$ l'ensemble des fautes dont l'absence est diagnostiquable en utilisant possiblement des décisions conditionnelles. Sous l'hypothèse qu'il est possible de partitionner l'ensemble des fautes E_F en $E_F = E_{F,F} \cup E_{F,NF}$, nous avons la définition suivante.

Définition 10 L est codiagnostiquable conditionnellement, dénoté par COND-CODIAG, par rapport à $E_{F,F}$ et $E_{F,NF}$ si :

1. L est COND-F-CODIAG par rapport à $E_{F,F}$; et
2. L is COND-NF-CODIAG par rapport à $E_{F,NF}$.

6.2. Propriétés de la Codiagnostiquabilité Conditionnelle

Théorème 5 Si L est codiagnostiquable par rapport à $E_{F,F}$ et $E_{F,NF}$, alors L est COND-F-CODIAG et COND-NF-CODIAG par rapport à tous les événements dans $E_{F,F} \cup E_{F,NF}$. Par contre, l'inverse n'est pas vrai en général.

Preuve: Nous démontrons que les fautes qui sont F-CODIAG et celles qui sont NF-CODIAG sont à la fois COND-F-CODIAG et COND-NF-CODIAG. Il y a quatre cas à considérer.

(i) D'après la définition de la propriété COND-F-CODIAG, nous avons directement que F-CODIAG implique COND-F-CODIAG.

(ii) F-CODIAG implique qu'il existe un entier k tel que pour chaque trace fautive s et suffixe t , avec $|t| \geq k$, il existe un site j pour lequel $\mathcal{E}_j(st)$ ne contient que des traces fautives. Par l'hypothèse (A2) de la Section 2, soit d le nombre maximum d'événements inobservables consécutifs dans toute trace de L . Dans le but de prouver la propriété de COND-NF-CODIAG, considérons la trace non-fautive uv , avec $|v| \geq nk(d+1)$. Alors v contient au moins nk événements observables (pas nécessairement par le même site). Puisqu'il y a n sites, il existe un site i qui observe au moins k de ces événements. Donc, nous avons : $P_i(v) \geq k$, $(\forall st \in \mathcal{E}_i(uv) : P_i(s) = P_i(u)$ et $P_i(t) = P_i(v) \geq k) |t| \geq k$. Si s est fautive, alors st doit être reconnue par un site j comme étant fautive, puisque L est F-CODIAG. Autrement dit, $\mathcal{E}_j(st)$ ne contient que des traces fautives. Par conséquent, le système est COND-NF-CODIAG.

(iii) et (iv) Ces deux cas portant sur NF-CODIAG impliquant COND-F-CODIAG et COND-NF-CODIAG sont prouvés de façon similaire.

L'exemple 6 qui suit démontre que COND-F-CODIAG n'implique pas la codiagnostiquabilité. Un exemple similaire peut être construit pour le case de COND-NF-CODIAG. ■

Exemple 6 Soit le système modélisé par le langage

$$\overline{c^*[a_1(b_2 + f) + b_1(f + a_2)]c^*}$$

où il y a deux sites et où $E_{o,1} = \{a_1, a_2, c\}$, $E_{o,2} = \{b_1, b_2, c\}$, et $E_{uo} = E_f = \{f\}$. Ce système n'est pas F-CODIAG puisque la trace fautive b_1fc^m a la même projection que la trace non-fautive c^m au site 1 et a la même projection que la trace non-fautive $b_1a_2c^m$ au site 2. D'autre part, ce système n'est pas NF-CODIAG puisque la trace non-fautive c^m a la même projection que la trace fautive b_1fc^m au site 1 et a la même projection que la trace fautive a_1fc^m au site 2. Par contre, ce système est COND-F-CODIAG. Si la trace fautive a_1fc^m a lieu, le site 1 va conclure que le système a

exécuté soit la trace fautive a_1fc^m ou la trace non-fautive $a_1b_2c^m$. Mais dans le cas de $a_1b_2c^m$, le site 1 conclut que le site 2 sera certain que $a_1b_2c^m$ a eu lieu. Donc le site 1 peut utiliser la décision F SI AUCUN NF après l'observation de a_1c , sachant que le site 2 dira NF s'il observe b_2c . On peut démontrer de façon similaire que la trace fautive b_1fc^m sera diagnostiquée. ■

Théorème 6 COND-F-CODIAG et COND-NF-CODIAG sont des propriétés incompatibles.

Preuve: Le système de l'exemple 6 est COND-F-CODIAG, mais il n'est pas COND-NF-CODIAG. Considérons la trace non-fautive c^m . Cette trace ne peut pas être distinguée de la trace b_1fc^m au site 1 ; le site 1 ne peut pas utiliser la décision NF SI AUCUN F puisque le site 2 ne peut pas distinguer b_1fc^m de la trace non-fautive $b_1a_2c^m$. On peut vérifier de façon similaire que c^m ne peut pas être diagnostiquée conditionnellement comme non-fautive pas la site 2.

Le preuve que COND-NF-CODIAG n'implique pas COND-F-DIAG est omise. ■

Nous terminons cette section avec deux résultats similaires à ceux du cas incondi-tionnel traité dans la Section 5.2. Les preuves de ces deux théorèmes sont similaires à leurs analogues inconditionnels (voir Théorèmes 3 et 4) et donc omises. La Fig. 2 résume les relations entre les différentes notions de diagnostiquabilité présentées dans cet article ; dans cette figure, le symbole \rightarrow signifie *implique*.

Théorème 7 COND-F-CODIAG ou COND-NF-CODIAG implique codiagnostiquabilité conditionnelle. Par contre, l'inverse n'est pas vrai en général.

Théorème 8 Codiagnostiquabilité conditionnelle par rapport à $E_{F,F}$ et $E_{F,NF}$ implique diagnostiquabilité (au sens centralisé, ou F-DIAG) pour tous les événements dans $E_{F,F} \cup E_{F,NF}$. Par contre, l'inverse n'est pas vrai en général.

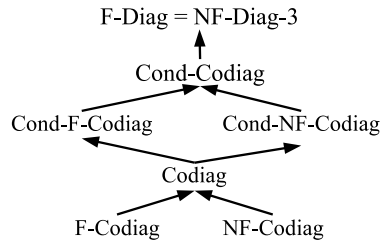


Figure 2. Relations entre les notions de codiagnostiquabilité

6.3. Discussion

On peut démontrer qu'il existe des algorithmes de complexité polynomiale pour vérifier les différentes notions de codiagnostiquabilité conditionnelle. La synthèse de diagnostiqueurs qui implémentent les décisions conditionnelles est un problème plus complexe et il n'est pas discuté dans cet article.

7. Conclusion

Le but de cet article est de démontrer comment élargir la classe de systèmes qui peuvent être diagnostiqués dans le cadre d'architectures décentralisées sans communication directe entre les diagnostiqueurs locaux et où il y a un coordonnateur qui peut implémenter des fonctions booléennes sans mémoire pour fusionner les décisions des diagnostiqueurs locaux. Dans ce contexte, nous avons vu comment l'emploi de différents types de décisions par rapport à la présence ou l'absence d'un événement de faute à diagnostiquer, F et NF, ainsi que l'emploi de décisions dites conditionnelles, F SI AUCUN NF et NF SI AUCUN F, permet effectivement d'atteindre le but fixé. Les résultats dans cet article améliorent donc ceux reliés au Protocole 3 dans (Debouk *et al.*, 2000) ainsi que les résultats récents de (Qiu *et al.*, 2004).

Remerciements

Cette recherche est subventionnée en partie par la US National Science Foundation (subvention CCR-0325571) et par la US Office of Naval Research (subvention N00014-03-1-0232). Les auteurs remercient les évaluateurs pour leurs commentaires pertinents.

8. Bibliographie

- Benveniste A., Haar S., Fabre E., Jard C., « Distributed and asynchronous discrete event systems diagnosis », *Proc. 41st IEEE Conf. on Decision and Control*, p. 3742-3747, December, 2003.
- Boel R. K., Jiroveanu G., « Distributed contextual diagnosis for very large systems », *Proc. of the 2004 International Workshop on Discrete Event Systems - WODES'04*, Reims, France, September, 2004.
- Boel R., van Schuppen J., « Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers », *Proc. of the 2002 International Workshop on Discrete Event Systems - WODES'02*, Zaragoza, Spain, October, 2002.
- Cassandras C. G., Lafortune S., *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.

- 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, n° 1-2, p. 33-86, January, 2000.
- Fabre E., Benveniste A., Jard C., Ricker L., Smith M., « Distributed state reconstruction for discrete event systems », *Proc. 39th IEEE Conf. on Decision and Control*, p. 2252-2257, December, 2000.
- Genc S., Lafortune S., « A distributed algorithm for on-line diagnosis of place-bordered Petri nets », *Proc. of 16th IFAC World Congress*, July, 2005.
- Lafortune S., Teneketzis D., Sampath M., Sengupta R., Sinnamohideen K., « Failure diagnosis of dynamic systems : An approach based on discrete event systems », *Proc. 2001 American Control Conf.*, p. 2058-2071, June, 2001.
- Lamperti G., Zanella M., *Diagnosis of Active Systems : Principles and Techniques*, Kluwer Academic Publishers, 2003.
- Qiu W., Kumar R., « Decentralized failure diagnosis of discrete event systems », *Proc. of the 2004 International Workshop on Discrete Event Systems - WODES'04*, Reims, France, September, 2004.
- Rudie K., Willems J. C., « The computational complexity of decentralized discrete-event control problems », *IEEE Trans. on Automat. Contr.*, vol. 40, n° 7, p. 1313-1318, July, 1995.
- Sampath M., Sengupta R., S. Lafortune K. S., Teneketzis D., « Diagnosability of discrete event systems », *IEEE Trans. on Automat. Contr.*, vol. 40, n° 9, p. 1555-1575, September, 1995.
- Sampath M., Sengupta R., S. Lafortune K. S., Teneketzis D., « Failure diagnosis using discrete event models », *IEEE Transactions on Control Systems Technology*, vol. 4, n° 2, p. 105-124, March, 1996.
- Sengupta R., « Diagnosis and communication in distributed systems », *Proc. of the 1998 International Workshop on Discrete Event Systems - WODES'98*, Cagliari, Italy, 1998.
- Sengupta R., Tripakis S., « Decentralized diagnosability of regular languages is undecidable », *Proc. 40th IEEE Conf. on Decision and Control*, p. 423-428, December, 2002.
- Su R., Wonham W., « Distributed diagnosis under global consistency », *Proc. 42nd IEEE Conf. on Decision and Control*, December, 2004.
- Su R., Wonham W., Kurien J., Koutsoukos X., « Distributed Diagnosis for Qualitative Systems », *Proc. of the 2002 International Workshop on Discrete Event Systems - WODES'02*, Zaragoza, Spain, p. 169-174, October, 2002.
- Wang Y., Lafortune S., *Decentralized Diagnosis of Discrete Event Systems : Architectures based on Unconditional and Conditional Decisions*, Technical Report n° CGR-05, College of Engineering Control Group Reports, University of Michigan, 2005.
- Yoo T.-S., Lafortune S., « A general architecture for decentralized supervisory control of discrete-event systems », *Discrete Event Dynamic Systems : Theory and Applications*, vol. 12, n° 3, p. 335-377, July, 2002a.
- Yoo T.-S., Lafortune S., « Polynomial-time verification of diagnosability of partially-observed discrete-event systems », *IEEE Trans. Automat. Contr.*, vol. 47, n° 9, p. 1491-1495, September, 2002b.
- Yoo T.-S., Lafortune S., « Decentralized supervisory control with conditional decisions : supervisor existence », *IEEE Trans. Automat. Contr.*, vol. 49, n° 11, p. 1886-1904, November, 2004.