

Une stratégie efficace pour une approche décentralisée du diagnostic de systèmes complexes

Yannick Pencolé

Marie-Odile Cordier

Laurence Rozé

IRISA

Campus de Beaulieu, 35042 Rennes cédex
{ypencole,cordier,roze}@irisa.fr

Résumé

Dans cet article, nous nous intéressons au problème du diagnostic de systèmes à événements discrets complexes tels que les réseaux de télécommunications. En raison de la taille importante de ce type de système, l'utilisation d'un modèle global est impossible. Nous proposons une approche en-ligne s'appuyant sur un modèle décentralisé et nous montrons comment préserver l'efficacité par l'utilisation d'une stratégie adéquate.

Mots Clef

Diagnostic à base de modèle, systèmes à événements discrets.

Abstract

We address the problem of diagnosing complex discrete-event systems such as telecommunication networks. Due to the size of the systems we are faced with, using a global model is clearly impossible. Our contribution is to propose an on-line approach which relies on a decentralized model and to show how the efficiency is preserved by using an adequate strategy.

Keywords

Model-based diagnosis, discrete-event systems.

1 Introduction

Nous nous intéressons au problème de la supervision de systèmes tels que les réseaux de télécommunications : ces systèmes à événements discrets sont complexes et de taille importante. Notre objectif est d'aider un opérateur de supervision à identifier les pannes du système, étant donné un flot continu d'alarmes reçues. Il existe deux approches classiques pour la surveillance de tels systèmes. Les techniques à base de connaissances associent directement un diagnostic à un ensemble de symptômes (par exemple les systèmes experts [9] ou les systèmes de reconnaissance

de scénarios [7]). Les techniques à base de modèle s'appuient sur un modèle de comportement du système pour produire le diagnostic ([13]). La faiblesse principale des approches à base de connaissances est le manque de généricité. Si le réseau est modifié, on doit reconsidérer l'ensemble de l'expertise et la plupart du temps effectuer une nouvelle acquisition de celle-ci afin d'adapter le système de diagnostic. Les techniques à base de modèles sont mieux adaptées aux systèmes évolutifs : une modification ne se répercute que sur le modèle de comportement du (ou des) composant(s) concerné(s). La mise à jour est plus localisée et donc moins coûteuse. Dans la suite, nous nous focalisons sur les approches à base de modèles.

Il existe de nombreuses approches pour le diagnostic de systèmes à événements discrets, que cela soit dans le domaine de l'IA ou dans le domaine du contrôle. Les systèmes concernés sont aussi bien des systèmes à variables continues qui, par échantillonnage, sont représentés par un modèle discret, que des systèmes "discrets par nature" (processus communicants par échange de messages).

La plupart de ces approches sont centralisées [15, 8, 14]. [16] propose une compilation de l'information nécessaire au diagnostic dans une structure de données appelée diagnostiqueur. Cette structure, de type automate, associe les pannes avec les observations afin d'effectuer un diagnostic en-ligne. Dans le cadre des réseaux de télécommunications, [14] propose une extension de cette approche qui est bien adaptée pour le diagnostic en-ligne. Néanmoins, une approche centralisée nécessite une information globale sur le système, ce qui n'est pas réaliste pour des systèmes tels que les réseaux de télécommunications. De plus, les réseaux sont des systèmes naturellement distribués donc il est plus facile de les modéliser avec un point de vue décentralisé. Suivant ce principe, [6] propose une approche s'appuyant sur un ensemble de diagnostiqueurs décentralisés. Le problème de cette approche est que

la construction d'un tel diagnostiqueur nécessite tout de même la connaissance d'un modèle global . [2] et [4] s'appuient eux aussi sur un modèle décentralisé, mais leurs approches concernent le problème de diagnostic hors-ligne.

Notre contribution est une approche s'appuyant sur un modèle décentralisé qui fournit un diagnostic en-ligne sur des systèmes à événements discrets de taille importante tels que les réseaux de télécommunications. L'idée est de calculer un diagnostic pour chaque composant du système (*diagnostic local*) puis de construire un diagnostic de tout le système (*diagnostic global*) à partir des diagnostics locaux. Afin de préserver l'efficacité du calcul du diagnostic global, un point-clé est alors la stratégie utilisée lors de l'opération de fusion des diagnostics locaux.

Cet article est organisé de la façon suivante. Dans un premier temps, nous décrivons un réseau de télécommunications simplifié qui nous servira d'exemple tout au long de cet article. Dans la section 3, nous introduisons le formalisme (automates communicants) que nous utilisons pour modéliser le système de façon décentralisée. La section 4 définit le problème de diagnostic décentralisé et en particulier l'opération de fusion qui permet le calcul du diagnostic global en fonction des diagnostics locaux. Nous présentons dans la section 5 la stratégie de fusion et en discutons l'efficacité. Nous concluons dans la section 6.

2 Exemple

Dans cette section, nous présentons un réseau de télécommunications simplifié (figure 1) que nous utiliserons comme exemple dans la suite de cet article. Ce

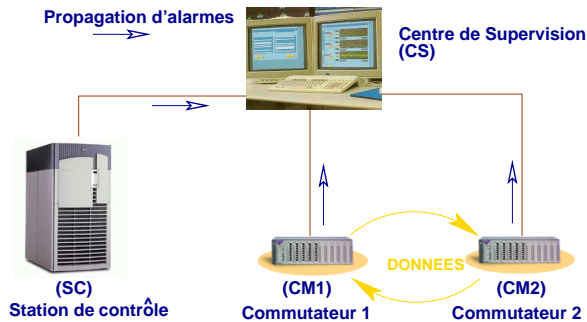


FIG. 1 – Exemple simplifié d'un réseau de télécommunications.

réseau est composé de deux commutateurs ($CM1$ et $CM2$) qui envoient et reçoivent les données circulant sur le réseau, d'une station de contrôle (SC) gérant ces commutateurs et d'un centre de supervision (CS). Les composants peuvent échanger des messages entre eux. Un composant peut aussi envoyer un message, appelé alors alarme, au superviseur. Le rôle du superviseur est de surveiller le système et en particulier de

repérer les occurrences de pannes par une analyse du flux d'alarmes envoyées par les composants SC , $CM1$ et $CM2$.

Nous supposons que les pannes ne peuvent survenir que sur les composants du réseau et que les connexions entre ces différents composants sont sûres. Une panne est définie par deux événements : le début de la panne et sa fin. Il existe des procédures automatiques de récupération de pannes et en général donc une panne se termine sans nécessiter l'intervention d'un opérateur.

Deux types de pannes peuvent se produire sur SC : la station peut se réinitialiser ou se bloquer. Ces pannes sont définies par un événement de début de panne ($SCbt$, $SCblc$) et par un événement de fin de panne ($SCfinbt$, $SCret$).

Lorsque une réinitialisation se déclenche sur SC , il y a émission d'une alarme $SCstop$. De plus, SC envoie les messages $CM1bt$ et $CM2bt$ aux commutateurs $CM1$ et $CM2$ pour qu'ils se réinitialisent eux-aussi. Quand SC commence à se bloquer, il émet aussi l'alarme $SCstop$. Si SC se réinitialise alors qu'il est bloqué, il n'émet pas de $SCstop$. Lorsque SC revient en bon état de fonctionnement (fin de réinitialisation $SCfinbt$ ou fin de blocage $SCret$), il émet l'alarme $SCok$.

Lorsque le commutateur $CM1$ se réinitialise (réinitialisation provoquée par SC), il émet l'alarme $CM1stop$. Lorsque $CM1$ se bloque ($CM1blc$), il émet l'alarme $CM1stop$ et provoque le blocage de $CM2$ par l'envoi de $CM2blc$. Si $CM1$ se réinitialise alors qu'il est bloqué, alors il n'émet pas d'alarme. Lors de son retour en bon fonctionnement (fin de réinitialisation $CM1finbt$ ou fin de blocage $CM1ret$), il émet une alarme $CM1ok$.

Le commutateur $CM2$ n'est sujet qu'à des pannes provoquées par les pannes des autres composants. Lorsque $CM2$ se réinitialise (réinitialisation provoquée par SC), il émet une alarme $CM2boot$. Lorsque $CM2$ se bloque (blocage provoqué par le blocage de $CM1$), il émet une alarme $CM2stop$. Lors de son retour en bon fonctionnement (fin de réinitialisation $CM2finbt$ ou fin de blocage $CM2ret$), il émet une alarme $CM2ok$.

3 Modèle décentralisé du système

Les systèmes que nous considérons sont des ensembles de composants qui interagissent les uns avec les autres. Comme dit précédemment, traiter ce genre de système à l'aide d'un modèle global n'est pas réaliste. Cette section explique comment le modèle du système est décrit de façon décentralisée à l'aide de modèles locaux. L'ensemble des modèles locaux décrit le comportement des composants et les interactions entre eux. Nous utilisons le formalisme des automates communicants pour décrire ce modèle.

Chaque composant peut réagir à des événements exogènes tels que les pannes en changeant d'états et en émettant des événements observables (appelés

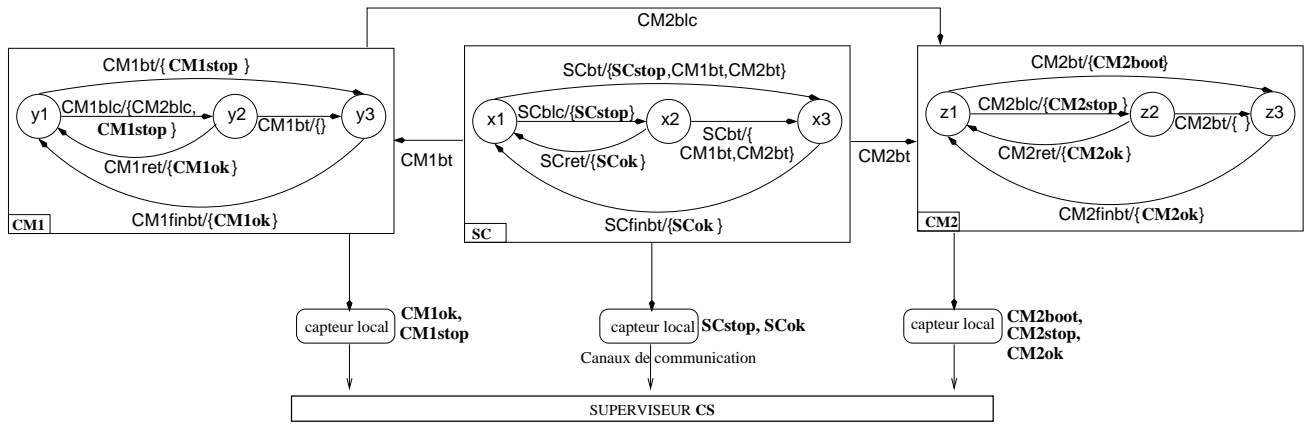


FIG. 2 – Modèle du système.

alarmes) au superviseur. Les composants interagissent entre eux par des échanges de messages (appelés *événements internes*) qui se produisent par exemple lorsque les pannes se propagent dans le système. Nous supposons qu'il n'y a pas de délais sur les messages échangés par les composants.

Le superviseur *CS* reçoit des alarmes de chaque composant *via* un canal de communication. Nous supposons que chaque canal de communication est associé à un unique composant. À chaque alarme est associé le composant qui l'a émise. Nous supposons également que les canaux de communications peuvent être représentés par des files FIFO et qu'il n'y a pas de synchronisation entre les différents canaux (le délai de propagation des alarmes sur un canal peut différer du délai sur un autre canal). Nous faisons donc les hypothèses suivantes :

Hypothèse 1 Soit o_1 et o_2 deux événements observables émis par un composant, o_1 et o_2 sont reçus par le superviseur dans l'ordre de leur émission par le composant.

Hypothèse 2 Soit o_1 et o_2 deux événements observables émis par deux composants différents, o_1 et o_2 peuvent ne pas être reçus par le superviseur dans l'ordre de leur émission.

Un composant i peut recevoir deux types d'événements : des événements exogènes (Σ_{exo}^i) tels que les événements de pannes et des événements internes ($\Sigma_{intrecu}^i$) émis par d'autres composants. Il peut émettre deux types d'événements : des événements observables (Σ_{obs}^i) à destination du superviseur (*via* son canal de communication) et des événements internes à destination des autres composants ($\Sigma_{intemis}^i$).

Définition 1 Le modèle d'un composant est un automate communicant Γ_i :

$$\Gamma_i = (\Sigma_{dec}^i, 2^{\Sigma_{emis}^i}, Q_i, E_i)$$

- Σ_{dec}^i est l'ensemble des événements déclencheurs ($\Sigma_{dec}^i = \Sigma_{exo}^i \cup \Sigma_{intrecu}^i$);
- Σ_{emis}^i est l'ensemble des événements émis par le composant ($\Sigma_{emis}^i = \Sigma_{obs}^i \cup \Sigma_{intemis}^i$);
- Q_i est l'ensemble des états du composant;
- $E_i \subseteq (Q_i \times \Sigma_{dec}^i \times 2^{\Sigma_{emis}^i} \times Q_i)$ est l'ensemble des transitions.

Le modèle du système est décrit de manière décentralisée par les modèles de ses composants.

Définition 2 Le modèle décentralisé Γ est décrit par l'ensemble des modèles locaux $\{\Gamma_1, \dots, \Gamma_n\}$, un ensemble d'événements exogènes (Σ_{exo}), un ensemble d'événements observables (Σ_{obs}) et un ensemble d'événements internes (Σ_{int}) tels que :

- $\{\Sigma_{obs}^1, \dots, \Sigma_{obs}^n\}$ est une partition de Σ_{obs} ;
- $\{\Sigma_{exo}^1, \dots, \Sigma_{exo}^n\}$ est une partition de Σ_{exo} ;
- $\{\Sigma_{intrecu}^1, \dots, \Sigma_{intrecu}^n\}$ et $\{\Sigma_{intemis}^1, \dots, \Sigma_{intemis}^n\}$ sont deux partitions de Σ_{int} ;
- $\forall e \in \Sigma_{int}, \exists ! \Gamma_i | e \in \Sigma_{intrecu}^i \wedge \exists ! \Gamma_j | e \in \Sigma_{intemis}^j \wedge i \neq j$.

Nous présentons sur la figure 2 le modèle du système décrit dans la section 2. Les événements observables sont les alarmes potentiellement reçues par le superviseur ($SCstop$, $SCok$, $CM1stop$, $CM1ok$, $CM2stop$, $CM2boot$, $CM2ok$). Les événements exogènes rassemblent les événements de pannes ($SCblc$, $SCret$, $SCbt$, $SCfinbt$, $CM1blc$, $CM1ret$, $CM1finbt$, $CM2ret$, $CM2finbt$). L'ensemble des événements internes ($CM1bt$, $CM2bt$) modélise la propagation liée à la réinitialisation de la station de contrôle. De même, $CM1blc$ modélise la propagation du blocage de $CM1$ sur $CM2$.

4 Diagnostic décentralisé

Établir un diagnostic consiste à calculer l'état de panne des composants du système et/ou les occurrences d'événements de pannes à partir d'une séquence

d'observations (dans notre cas la séquence d'alarmes reçues par le superviseur). Comme chez plusieurs auteurs [2, 5, 3], nous considérons que la tâche de diagnostic consiste, pour les systèmes dynamiques, à trouver les trajectoires (séquence d'états et transitions) expliquant les observations. À partir de ces trajectoires, il est facile d'extraire le diagnostic lui-même. Dans un cas simple, on peut être uniquement intéressé par l'ensemble des composants en pannes mais dans le cadre de la surveillance d'un système, l'opérateur est souvent intéressé par une explication plus détaillée de ce qui s'est produit. En particulier, il est intéressé par la séquence d'événements de pannes exogènes qui peut expliquer l'ensemble des alarmes reçues et l'ensemble des états courants possibles du système.

L'idée est d'utiliser une approche décentralisée en calculant un diagnostic pour chaque composant (*diagnostic local*) puis en produisant le diagnostic de tout le système (*diagnostic global*) à partir des diagnostics locaux. Dans la suite, nous définissons formellement ces notions et expliquons la construction du diagnostic global par *composition* des diagnostics locaux.

Nous considérons que l'ensemble des séquences d'observations $\mathcal{O} = \{\mathcal{O}_{\Gamma_1}, \dots, \mathcal{O}_{\Gamma_n}\}$, $\mathcal{O}_{\Gamma_i} \in (\Sigma_{obs}^i)^*$ où chaque \mathcal{O}_{Γ_i} est la séquence d'observations reçues des composants Γ_i durant une fenêtre temporelle Δ_t ¹. Au début de la fenêtre temporelle, les états initiaux X_{init}^Γ de Γ sont connus et décrits par les états initiaux de chaque composant $\{X_{init}^{\Gamma_1}, \dots, X_{init}^{\Gamma_n}\}$.

4.1 Diagnostic local

Étant donné un modèle de composant Γ_i et un ensemble d'états initiaux $X_{init}^{\Gamma_i}$, un *diagnostic local* Δ_{Γ_i} décrit le sous-ensemble de trajectoires de Γ_i qui, à partir d'un état de $X_{init}^{\Gamma_i}$, expliquent la séquence d'observations locales \mathcal{O}_{Γ_i} , et donc telles que la projection des trajectoires sur les événements observables de Γ_i correspond à \mathcal{O}_{Γ_i} . Nous proposons de représenter un diagnostic local par un automate communicant $\Delta_{\Gamma_i}(X_{init}^{\Gamma_i}, \mathcal{O}_{\Gamma_i})$ (notation simplifiée Δ_{Γ_i}). En comparaison avec l'automate Γ_i , la principale différence syntaxique est que chaque état Q_{Δ_i} de cet automate est un couple $(s_{\Gamma_i}, \mathcal{E}_i)$ où $s_{\Gamma_i} \in Q_i$ est un état du composant et \mathcal{E}_i est la sous-séquence préfixe de \mathcal{O}_{Γ_i} expliquée dans cet état. Les *états initiaux* de Δ_{Γ_i} sont du type (x, ϵ) où $x \in X_{init}^{\Gamma_i}$ et ϵ est la séquence vide. Les *états finals* de Δ_{Γ_i} sont du type $(x, \mathcal{O}_{\Gamma_i})$. Les états finals du diagnostic représentent les états courants du composant expliquant toute la séquence d'observations locales. Comme dans Γ_i , les transitions sont étiquetées avec des événements reçus et émis par le composant (événements exogènes, internes et observables).

Définition 3 (Diagnostic local) *Le diagnostic lo-*

¹La façon dont les fenêtres temporelles successives sont prises en compte ne fait pas l'objet de cet article et est brièvement commentée dans la conclusion.

cal $\Delta_{\Gamma_i}(X_{init}^{\Gamma_i}, \mathcal{O}_{\Gamma_i})$ de Γ_i selon la séquence d'observations \mathcal{O}_{Γ_i} est un automate communicant :

$$(\Sigma_{dec}^i, 2^{(\Sigma_{emis}^i)}, Q_{\Delta_i}, E_{\Delta_i})$$

où

- Σ_{dec}^i est l'ensemble des événements déclencheurs ($\Sigma_{dec}^i = \Sigma_{exo}^i \cup \Sigma_{intrecu}^i$);
- Σ_{emis}^i est l'ensemble des événements émis par le composant ($\Sigma_{emis}^i = \Sigma_{obs}^i \cup \Sigma_{intemis}^i$);
- Q_{Δ_i} est l'ensemble des états ($Q_{\Delta_i} \subseteq Q_i \times (\Sigma_{obs}^i)^*$);
- $E_{\Delta_i} \subseteq (Q_{\Delta_i} \times \Sigma_{dec}^i \times 2^{(\Sigma_{emis}^i)} \times Q_{\Delta_i})$ est l'ensemble des transitions.

La figure 3 présente le diagnostic local de la station de contrôle lorsque $X_{init}^{SC} = \{x1\}$ et $\mathcal{O}_{SC} = [SCstop, SCok]$. Il contient 3 trajectoires et affirme qu'un seul état du composant est possible après ces observations : l'état x_1 .

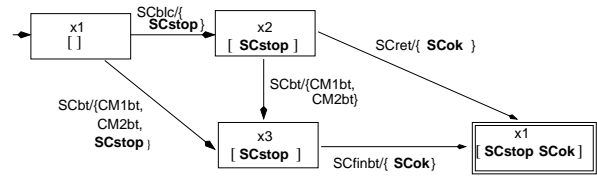


FIG. 3 – Diagnostic local : $\Delta_{SC}(x1, [SCstop, SCok])$.

La façon de calculer un diagnostic local n'est pas le sujet de cet article. Dans notre cas, nous utilisons une approche diagnostiqueur détaillée dans [11][10]. Dans la section suivante, nous nous focalisons sur la construction du diagnostic global à partir des diagnostics locaux.

4.2 Diagnostic global : fusion de diagnostics locaux

Nous avons défini un diagnostic local comme un automate communicant qui décrit les trajectoires possibles d'un composant expliquant une séquence d'observations locales.

De même, à partir du modèle du système Γ et d'un ensemble d'états initiaux X_{init}^Γ , le diagnostic global est un automate communicant $\Delta_\Gamma(X_{init}^\Gamma, \mathcal{O})$ qui décrit les trajectoires possibles de Γ expliquant le flux d'observations \mathcal{O} reçues par le superviseur. En raison des hypothèses 1 et 2, \mathcal{O} est complètement décrit par l'ensemble des séquences $\{\mathcal{O}_{\Gamma_1}, \dots, \mathcal{O}_{\Gamma_n}\}$. Les trajectoires sont donc celles telles que leurs projections sur les événements observables de chacun des Γ_i correspondent à chacune des séquences \mathcal{O}_{Γ_i} .

Le diagnostic global est un automate communicant $\Delta_\Gamma(X_{init}^\Gamma, \mathcal{O})$ (notation simplifiée Δ_Γ). Les états Q_Δ de cet automate sont des couples (s_Γ, \mathcal{E}) où $s_\Gamma \in Q_1 \times \dots \times Q_n$ est un état du système et $\mathcal{E} = \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ où \mathcal{E}_i est le préfixe de la séquence \mathcal{O}_{Γ_i} expliquée

dans cet état. Les états initiaux de Δ_Γ sont ceux correspondant à X_{init}^Γ et les états finals sont tels que $\forall i \mathcal{E}_i = \mathcal{O}_{\Gamma_i}$. Ainsi les états finals représentent les états courants des composants du système expliquant l'ensemble des observations. Les transitions sont étiquetées par des événements exogènes ($\Sigma_{dec}^\Gamma = \Sigma_{exo}$) et par des événements observés ($\Sigma_{émis}^\Gamma = \Sigma_{obs}$).

Définition 4 (Diagnostic global) *Le diagnostic global $\Delta_\Gamma(X_{init}^\Gamma, \mathcal{O})$ de Γ selon \mathcal{O} est un automate communicant : $(\Sigma_{dec}, 2^{\Sigma_{émis}}, Q_\Delta, E_\Delta)$ où*

- Σ_{dec} est l'ensemble des événements déclencheurs ($\Sigma_{dec} = \Sigma_{exo}$);
- $\Sigma_{émis}$ est l'ensemble des événements émis ($\Sigma_{émis} = \Sigma_{obs}$);
- Q_Δ est l'ensemble des états du diagnostic;
- $E_\Delta \subseteq (Q_\Delta \times \Sigma_{dec} \times 2^{\Sigma_{émis}} \times Q_\Delta)$ est l'ensemble des transitions.

La figure 4 présente le diagnostic global du modèle décrit dans la figure 2 pour $X_{init}^\Gamma = \{(x1, y1, z1)\}$ et $\mathcal{O} = \{[SCstop, SCok], [CM1stop], [CM2boot]\}$. Il ne contient qu'un unique état final, signifiant qu'après l'observation de \mathcal{O} , le système est nécessairement dans l'état $(x1, y3, z3)$. Ce diagnostic représente 2 trajectoires pouvant expliquer \mathcal{O} .

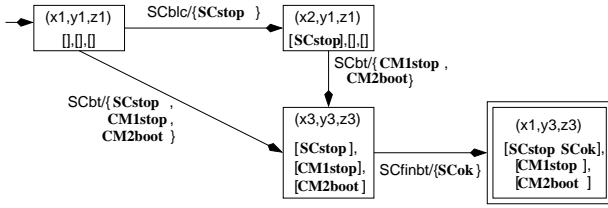


FIG. 4 - Diagnostic global du système : $\Delta_\Gamma(\{(x1, y1, z1)\}, \{[SCstop, SCok], [CM1stop], [CM2boot]\})$.

Dans une approche décentralisée, le modèle Γ du système n'est pas donné explicitement mais représenté par l'ensemble des modèles de ses composants. Le diagnostic global n'est donc pas construit directement à partir du modèle global mais il est construit à partir des diagnostics locaux. On a

$$\Delta_\Gamma(X_{init}^\Gamma, \mathcal{O}) = \bigodot_{i=1}^n \Delta_{\Gamma_i}(X_{init}^{\Gamma_i}, \mathcal{O}_{\Gamma_i})$$

où \odot est une opération de composition basée sur un produit synchronisé des automates représentant les diagnostics locaux [1], la synchronisation s'effectuant sur les échanges de messages entre diagnostics [11].

Le diagnostic global de la figure 4 est ainsi obtenu par l'opération $\Delta_{SC}(x1, [SCstop, SCok]) \odot \Delta_{CM1}(y1, [CM1stop]) \odot \Delta_{CM2}(z1, [CM2boot])$.

Le point intéressant dans la définition du diagnostic donnée ci-dessus est que l'opération de composition

des automates communicants est commutative et associative. Le diagnostic global peut être calculé de plusieurs façons selon l'ordre dans lequel les diagnostics sont progressivement fusionnés (composés). L'efficacité du calcul va dépendre directement de ce point clé.

Dans la suite, nous nous focalisons sur la stratégie de fusion et nous proposons de l'établir en fonction des interactions existant entre les diagnostics.

5 Stratégie de fusion

Dans cette section, nous considérons le problème de l'efficacité du calcul du diagnostic global à partir des diagnostics locaux et proposons une stratégie pour améliorer le coût des fusions successives. Cette stratégie consiste à faire au plus tôt les fusions des composants qui ont le plus d'interactions entre eux. L'existence d'interactions entre composants est connue par une analyse de leurs diagnostics locaux.

5.1 Définitions

Les trajectoires d'un diagnostic local Δ_{Γ_i} qui font l'hypothèse d'interactions entre le composant diagnostiqué (Γ_i) et d'autres composants sont celles comprenant des événements de type échanges de messages internes (Σ_{int}).

Notation 1 ($\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)$) *On note $\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)$ l'ensemble des événements du composant Γ_i qui sont supposés avoir été échangés avec le composant Γ_j d'après le diagnostic local de Γ_i (Δ_{Γ_i}).*

Définition 5 (Trajectoire impossible) *Une trajectoire du diagnostic local Δ_{Γ_i} est dite impossible ssi il existe un composant Γ_j et une transition dans la trajectoire qui suppose l'échange d'un événement $e \in \mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)$ tel que $e \notin \mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i)$.*

Un diagnostic local Δ_{Γ_i} peut émettre l'hypothèse qu'un événement e a été échangé entre Γ_i et Γ_j ($e \in \mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)$) alors que Δ_{Γ_j} n'a aucune hypothèse telle que e appartient à $\mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i)$. Les trajectoires de Δ_{Γ_i} qui supposent l'échange de e avec Δ_{Γ_j} sont donc impossibles : elles ne peuvent pas faire partie d'une trajectoire du diagnostic global et peuvent donc être éliminées avant toute fusion de diagnostic.

Nous appelons *diagnostic épuré* de Γ_i le diagnostic obtenu après avoir éliminé de Δ_{Γ_i} toutes les trajectoires impossibles ² et nous le notons Δ'_{Γ_i} .

Définition 6 (Composants interagissants) *On dit que Γ_i et Γ_j interagissent ssi $\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j) \cap \mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i) \neq \emptyset$.*

On peut en déduire que Γ_i et Γ_j interagissent ssi leurs diagnostics épurés sont tels que $\mathcal{I}_{\Delta'_{\Gamma_i}}(\Gamma_j) = \mathcal{I}_{\Delta'_{\Gamma_j}}(\Gamma_i) = \mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j) \cap \mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i) \neq \emptyset$. Nous dirons

²On suppose que le modèle du système est complet et donc qu'il existe toujours un diagnostic épuré.

que Γ_i et Γ_j sont *k-interagissent* ssi ils interagissent et $|\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)| = |\mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i)| = |\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j) \cap \mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i)| = k$. Par extension de la définition, on dit que deux ensembles distincts de composants γ_1, γ_2 sont *k-interagissent* ssi

$$\sum_{(\Gamma_i, \Gamma_j) | \Gamma_i \in \gamma_1, \Gamma_j \in \gamma_2} |\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)| = k, k > 0.$$

5.2 Principes et algorithme

L'opération de fusion (ou composition) notée \odot est une opération commutative et associative, aussi nous pouvons composer les diagnostics dans n'importe quel ordre. Afin d'être efficace, nous allons adopter les deux principes suivants (détaillé dans l'algorithme 1).

1. Détecter et éliminer les trajectoires impossibles.

Dans une mise en œuvre classique de la composition d'automates, les trajectoires impossibles sont gérées de la même façon que les autres, à savoir en essayant de les composer jusqu'à détecter un succès ou un échec. Les trajectoires augmentent inutilement le coût de l'opération de composition. On va donc les éliminer avant l'application de toute opération de composition. L'idée est de calculer les ensembles $\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)$ ³ et de détecter les événements $e \in \mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j)$ tels que $e \notin \mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i)$ (contenu dans *EvtPasBon*, voir l'algorithme). Les trajectoires impossibles sont alors éliminées en utilisant un algorithme adapté (*ElimTrajImpos*). On obtient donc le diagnostic épuré Δ'_{Γ_i} et les ensembles $\{\mathcal{I}_{\Delta'_{\Gamma_i}}(\Gamma_j), j \in \{1, \dots, n\}, j \neq i\}$.

2. Composer les diagnostics locaux de composants qui interagissent en premier.

Composer deux diagnostics locaux épurés dont les composants n'interagissent pas revient à calculer leur produit cartésien (pas d'interactions entre leurs trajectoires). Nous voulons éviter ce type de calcul. Pour connaître les composants interagissants avec Γ_i ($kInter(\Gamma_i)$), on utilise les ensembles d'événements $\mathcal{I}_{\Delta'_{\Gamma_i}}(\Gamma_j)$. (Γ_j, k) appartient à $kInter(\Gamma_i)$ si Γ_i et Γ_j *k-interagissent*. Ensuite, on démarre la composition en choisissant une partition des diagnostics locaux en s'appuyant sur les ensembles $kInter(\Gamma_i)$ (*ChoisirPartitionAvecInter*). Nous favorisons une partition de diagnostics telles que :

- tout élément de la partition contient au plus deux diagnostics;
- tout élément de la partition contient deux diagnostics dont les composants *k-interagissent* avec *k* aussi grand que possible.

³Une façon efficace de le faire est d'utiliser l'approche diagnostiqueur proposée dans [11].

Algorithme 1 Calcul du diagnostic global

Entrée : $\{\Delta_{\Gamma_i}, i \in \{1, \dots, n\}\}$

Entrée : $\{\mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j), i, j \in \{1, \dots, n\}, j \neq i\}$

{1– Élimination des trajectoires impossibles}

pour tout $i \in \{1, \dots, n\}$ **faire**

$\Delta_{tmp} \leftarrow \Delta_{\Gamma_i}$

pour tout $j \in \{1, \dots, n\}, j \neq i$ **faire**

$EvtPasBon \leftarrow \mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j) - \mathcal{I}_{\Delta_{\Gamma_j}}(\Gamma_i)$

$\{EvtPasBon : \text{événements produisant des trajectoires impossibles dans } \Delta_{\Gamma_i} \text{ selon } \Delta_{\Gamma_j}.\}$

$\mathcal{I}_{\Delta'_{\Gamma_i}}(\Gamma_j) \leftarrow \mathcal{I}_{\Delta_{\Gamma_i}}(\Gamma_j) - EvtPasBon$

$\Delta_{tmp} \leftarrow ElimTrajImpos(\Delta_{tmp}, EvtPasBon)$

$\{\Delta_{tmp} \text{ n'a plus de trajectoires impossibles selon } \Delta_{\Gamma_j}.\}$

fin pour

$\Delta'_{\Gamma_i} \leftarrow \Delta_{tmp}$

fin pour

{2– Recherche de composants interagissants}

pour tout $i \in \{1, \dots, n\}$ **faire**

$kInter(\Gamma_i) \leftarrow \{(\Gamma_j, |\mathcal{I}_{\Delta'_{\Gamma_i}}(\Gamma_j)|) \mid \mathcal{I}_{\Delta'_{\Gamma_i}}(\Gamma_j) \neq \emptyset\}$

fin pour

{3– Stratégie de composition en parallèle}

$\mathcal{D} \leftarrow \{\Delta'_{\Gamma_i}, i \in \{1, \dots, n\}\}$

$\mathcal{M} \leftarrow \{kInter(\Gamma_i), i \in \{1, \dots, n\}\}$

tant que $\mathcal{M} \neq \emptyset$ **faire**

$\pi_{\mathcal{D}} \leftarrow ChoisirPartitionAvecInter(\mathcal{D}, \mathcal{M})$

$\mathcal{D} \leftarrow ComposeEnParallele(\pi_{\mathcal{D}})$

$\mathcal{M} \leftarrow MiseAJourInter(\mathcal{M}, \mathcal{D})$

fin tant que

L'ensemble \mathcal{D} représente le diagnostic global.

Une fois la partition de diagnostic choisie, on compose les diagnostics de chaque élément de la partition entre eux en parallèle (*ComposeEnParallele*). Nous obtenons un nouvel ensemble de diagnostics (un diagnostic pour chaque élément de partition). Nous mettons à jour les relations d'interaction en fonction du nouvel ensemble de diagnostics (*MiseAJourInter*). Ensuite, il suffit de réitérer en choisissant une nouvelle partition de diagnostics. Le résultat final est un ensemble de diagnostics. Chaque diagnostic est associé à un sous-ensemble de composants et affirme qu'il n'y a pas d'interaction entre ce sous-ensemble et un autre. Autrement dit, selon les diagnostics obtenus, chaque sous-ensemble a un

comportement indépendant des autres. Le diagnostic global est donc représenté implicitement par cet ensemble de diagnostics, il est inutile de continuer à fusionner.

5.3 Exemple

On suppose que $X_{init}^\Gamma = \{\{x1\}, \{y1\}, \{z1\}\}$ et que $\mathcal{O} = \{[SCstop, SCok], [CM1stop], [CM2boot]\}$ et on veut $\Delta_\Gamma(X_{init}^\Gamma, \mathcal{O})$ (voir figure 4). Les diagnostics locaux sont les suivants :

- $\Delta_{SC}(\{x1\}, [SCstop, SCok])$: 4 états, 5 transitions.
 $\mathcal{I}_{\Delta_{SC}}(CM1) = \{CM1bt\}$, $\mathcal{I}_{\Delta_{SC}}(CM2) = \{CM2bt\}$
 (cf figure 3).
- $\Delta_{CM1}(\{y1\}, [CM1stop])$: 3 états, 3 transitions.
 $\mathcal{I}_{\Delta_{CM1}}(SC) = \{CM1bt\}$, $\mathcal{I}_{\Delta_{CM1}}(CM2) = \{CM2blc\}$
 (cf figure 5).
- $\Delta_{CM2}(\{z1\}, [CM2boot])$: 2 états, 1 transitions.
 $\mathcal{I}_{\Delta_{CM2}}(SC) = \{CM2bt\}$, $\mathcal{I}_{\Delta_{CM2}}(CM1) = \emptyset$ (cf figure 5).

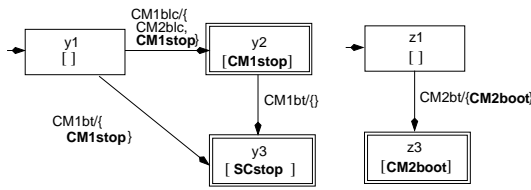


FIG. 5 – Diagnostics locaux de $CM1$ (gauche) et de $CM2$ (droite).

Le diagnostic local de $CM1$ affirme que $CM2blc$ a pu être échangé entre $CM1$ et $CM2$ ($\mathcal{I}_{\Delta_{CM1}}(CM2) = \{CM2blc\}$), par contre le diagnostic local de $CM2$ affirme que $CM2blc$ n'a pas été échangé ($\mathcal{I}_{\Delta_{CM2}}(CM1) = \emptyset$). Ainsi, les trajectoires de $\Delta_{CM1}(y1, [CM1stop])$ contenant $CM2blc$ sont impossibles. On élimine ces trajectoires et l'on obtient le diagnostic épuré $\Delta'_{CM1}(y1, [CM1stop])$ (cf figure 6).

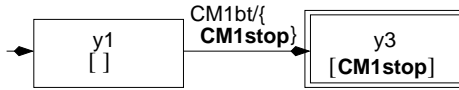


FIG. 6 – Diagnostic épuré de $CM1$.

Ce diagnostic contient 2 états et 1 transition. Les autres diagnostics restent inchangés car on n'y a pas détecté de trajectoires impossibles.

Ensuite, on compose l'ensemble de diagnostic. On ne compose pas les diagnostics de $CM1$ et de $CM2$ car $CM1$ et $CM2$ n'interagissent pas selon les diagnostics ($kInter(CM1) = \{(SC, 1)\}$ et $kInter(CM2) = \{(SC, 1)\}$). Dans cet exemple, SC et $CM1$ sont 1-interagissants comme SC et $CM2$, aussi nous avons deux possibilités ; par exemple, on peut appliquer la composition sur les diagnostics de SC et de $CM1$

en premier. Ensuite, après mise à jour, on s'aperçoit que $\{SC, CM1\}$ et $\{CM2\}$ sont 1-interagissants. Dans une seconde étape, on applique donc la composition entre le diagnostic résultat de la première composition et le diagnostic de $CM2$. Nous obtenons au final un ensemble de diagnostics ne contenant qu'un seul élément : le diagnostic global.

5.4 Résultats

Dans cette section, nous présentons une étude comparative de différentes stratégies de fusion sur un exemple. Cette étude se fonde sur un réseau plus complexe que celui présenté en section 2. Néanmoins, les équipements de ce réseau ont un comportement identique à ceux déjà présentés. Ce réseau dispose de deux stations de contrôle ($SC1$ et $SC2$) gérant dix commutateurs (cf figure 7). $SC1$ gère les commutateurs de 1 à 5 et $SC2$ ceux de 6 à 10.

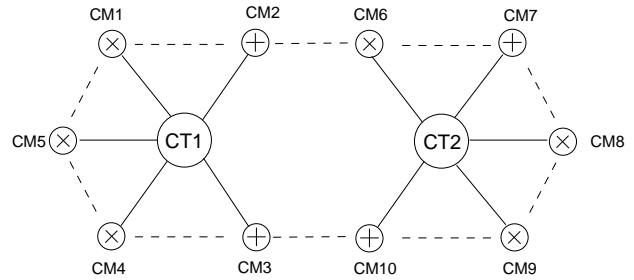


FIG. 7 – Comparaison des temps d'exécution pour quatre stratégies différentes.

Les commutateurs représentés par \otimes ont un comportement identique au commutateur $CM1$ de la section 2. De même, les commutateurs représentés par \oplus ont un comportement identique au commutateur $CM2$ de la section 2. En particulier, tout commutateur \otimes peut bloquer tout commutateur \oplus voisin.

Pour cette expérience, nous avons traité 26 alarmes (cf table 1). La table 2 présente la stratégie de fusion utilisée dans ce cas et déterminée par l'algorithme 1.

$SC1$	$SCstop$
$SC2$	$SCstop$
$CM1$	$CMstop, CMok$
$CM2$	$CMboot, CMok$
$CM3$	$CMstop, CMok$
$CM4$	$CMstop, CMok$
$CM5$	$CMstop, CMok, CMstop$
$CM6$	$CMstop$
$CM7$	$CMboot, CMok, CMstop$
$CM8$	$CMstop, CMok, CMstop$
$CM9$	$CMstop, CMok, CMstop$
$CM10$	$CMboot, CMok, CMstop$

TAB. 1 – Journal d'alarmes.

Dans cet exemple, l'algorithme a déterminé en particulier qu'il n'y avait pas d'interaction entre les commutateurs $CM6$ et $CM2$ (pas de blocage de $CM6$). Cette absence d'interaction a conduit à la construction de diagnostic sur deux sous-groupes indépendants du système. Dans la table 2, nous présentons les différentes opérations effectuées. Les étapes de 1 à 5 peuvent être effectuées en parallèle, ainsi que les étapes 6 à 7 et enfin les étapes 8 à 9. Les diagnostics des étapes 9 et 10 constituent le résultat du diagnostic. Ils informent en particulier que les stations de contrôle sont en cours de réinitialisation et que $CM5$ est soit bloqué, soit en cours de réinitialisation.

Étapes	Fusion effectuée
1	$\Delta_{SC1} \odot \Delta_{CM1} \rightarrow \Delta_{SC1,CM1}$
2	$\Delta_{SC2} \odot \Delta_{CM6} \rightarrow \Delta_{SC2,CM6}$
3	$\Delta_{CM3} \odot \Delta_{CM4} \rightarrow \Delta_{CM3,CM4}$
4	$\Delta_{CM7} \odot \Delta_{CM8} \rightarrow \Delta_{CM7,CM8}$
5	$\Delta_{CM9} \odot \Delta_{CM10} \rightarrow \Delta_{CM9,CM10}$
6	$\Delta_{SC1,CM1} \odot \Delta_{CM3,CM4}$ $\rightarrow \Delta_{SC1,CM1,CM3,CM4}$
7	$\Delta_{SC2,CM6} \odot \Delta_{CM7,CM8}$ $\rightarrow \Delta_{SC2,CM6,CM7,CM8}$
8	$\Delta_{SC1,CM1,CM3,CM4} \odot \Delta_{CM2}$ $\rightarrow \Delta_{SC1,CM1,CM2,CM3,CM4}$
9	$\Delta_{SC2,CM6,CM7,CM8} \odot \Delta_{CM9,CM10}$ $\rightarrow \Delta_{SC2,CM6,CM7,CM8,CM9,CM10}$
10	$\Delta_{SC1,CM1,CM2,CM3,CM4} \odot \Delta_{CM5}$ $\rightarrow \Delta_{SC1,CM1,CM2,CM3,CM4,CM5}$

TAB. 2 – Stratégie utilisée.

Afin de montrer l'intérêt de la stratégie décrite par l'algorithme 1, nous l'avons comparée avec d'autres stratégies de fusion. La figure 8 présente les temps d'exécution pour quatre stratégies. Pour chaque étape, nous présentons le temps d'exécution dans cette étape cumulé avec ceux des étapes précédentes.

La stratégie 1 est celle décrite dans le tableau 2. Dans la stratégie 2, les fusions produites sont identiques mais on n'a pas appliqué au préalable l'élimination de trajectoires impossibles. Dans la stratégie 3, l'élimination des trajectoires a été effectuée mais on a privilégié la fusion de diagnostics non interagissants. Quant à la stratégie 4, elle ne suit aucun des deux principes : cette stratégie montre l'effet cumulatif des principes adoptés.

6 Conclusion

Cet article décrit une approche de diagnostic décentralisée adaptée aux systèmes à événements discrets de grande taille tels que les réseaux de télécommunications. Utiliser un modèle global de ce genre de système est clairement impossible. L'idée est donc de calculer des diagnostics locaux à partir

de modèles locaux et de construire le diagnostic global par fusion de ces résultats locaux. L'objectif est de préserver l'efficacité du calcul pour effectuer du diagnostic en-ligne. Deux points sont à considérer. Le premier est l'ordre dans lequel les fusions successives sont effectuées. Nous proposons dans ce papier une stratégie favorisant la fusion de diagnostics de composants interagissants. Cette stratégie peut être comparée au plan de reconstruction proposé par [2]. Le principal avantage de notre proposition est que les interactions sont recherchées par une analyse en ligne des diagnostics, la stratégie est donc dynamique et adaptable. Le second point important est le choix de la taille de la fenêtre temporelle. Ce point n'a pas été développé dans cet article où nous considérons une séquence d'observations d'une fenêtre temporelle donnée. Un algorithme incrémental capable de prendre en compte des observations sur des fenêtres temporelles successives est développé dans [12]. La taille optimale de la fenêtre temporelle y est discutée, sachant qu'elle dépend du système étudié.

Références

- [1] A. Arnold. *Systèmes de transitions finis et sémantique des processus communicants*. Masson, 1992.
- [2] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella. Diagnosis of large active systems. *Artificial Intelligence*, 110:135–183, 1999.
- [3] C. Barral, S. McIlraith, and T. C. Son. Formulating diagnostic problem solving using an action language with narratives and sensing. In *Proceedings of KR'2000*, pages 311–322, Cambridge, 2000.
- [4] L. Console, C. Picardi, and M. Ribaud. Diagnosis and diagnosability analysis using pepa. In *Proceedings of the European Conference on Artificial Intelligence (ECAI'2000)*, pages 131–135, Berlin, 2000.
- [5] M-O. Cordier and S. Thiébaux. Event-based diagnosis for evolutive systems. In *Proceedings of the International Workshop on Principles of Diagnosis (DX-94)*, pages 64–69, New Palz (NY), October 1994.
- [6] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems*, 10(1-2):33–86, 2000.
- [7] C. Dousson, P. Gaborit, and M. Ghallab. Situation recognition: representation and algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 166–172, Chambéry, France, 1993.
- [8] J. Lunze. Discrete-event modelling and diagnosis of quantized dynamical systems. In *Proceedings*

Comparaison de stratégies

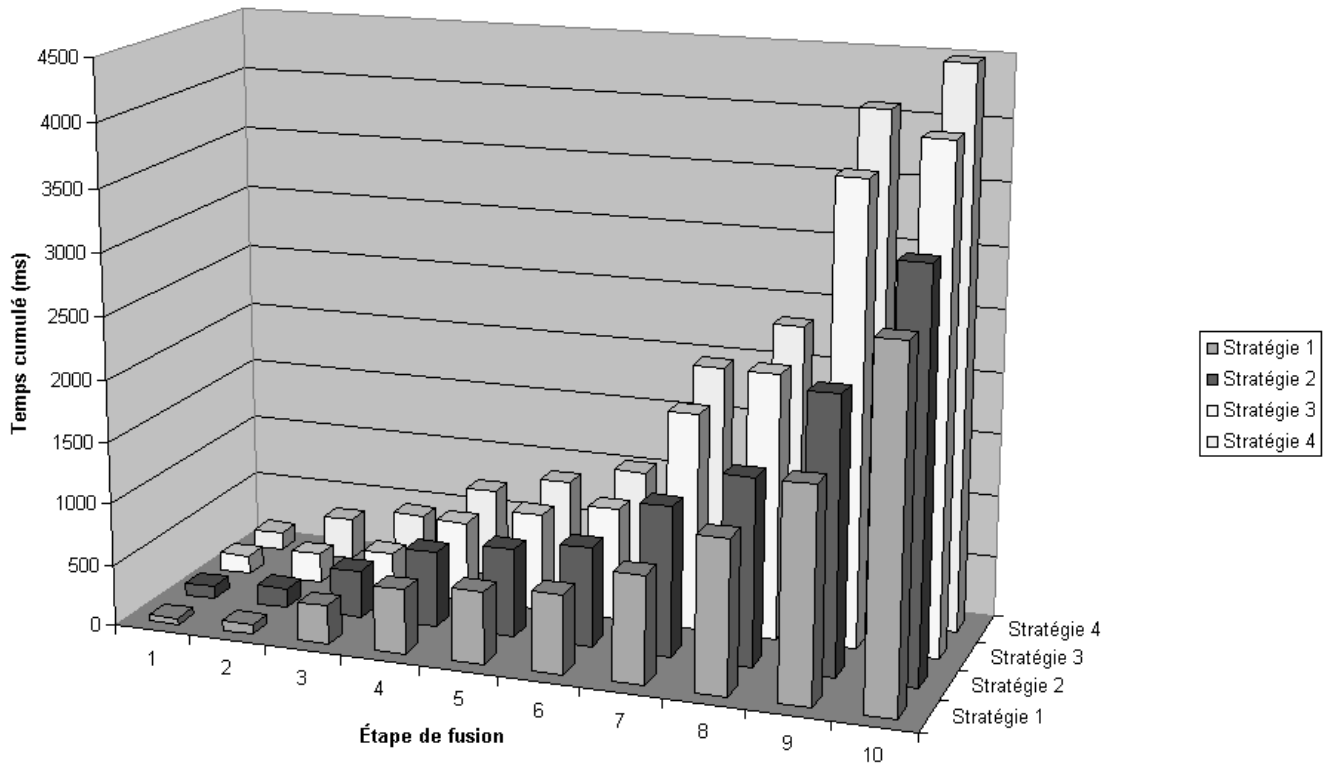


FIG. 8 – Comparaison des temps d'exécution pour quatre stratégies différentes.

of the *International Workshop on Principles of Diagnosis (DX'99)*, pages 147–154, Loch Awe, 1999.

- [9] D. Niebur. Expert systems for power system control in western europe. *Proceedings of the IEEE Symposium on Intelligent Control*, pages 112–119, 1990.
- [10] Y. Pencolé. Approche diagnostiqueur décentralisé : application aux réseaux de télécommunications. In *Actes de RJCIA'2000 (5èmes Rencontres nationales des Jeunes Chercheurs en Intelligence Artificielle)*, Lyon, France, 2000.
- [11] Y. Pencolé. Decentralized diagnoser approach: application to telecommunication networks. In *Proceedings of the International Workshop on Principles of Diagnosis (DX'00)*, pages 185–192, Morelia, Mexico, 2000.
- [12] Y. Pencolé, M-O. Cordier, and L. Rozé. Incremental decentralized diagnosis approach for the supervision of a telecommunication network. In *Proceedings of the International Workshop on Principles of Diagnosis (DX'01)*, pages 151–158, Sansicario, Italy, 2001.
- [13] M. Riese. Diagnosis of extended finite automata as a dynamic constraint satisfaction problem. In *Proceedings of the International Workshop on Principles of Diagnosis (DX'93)*, pages 60–73, Aberystwyth, UK, 1993.
- [14] L. Rozé and M-O. Cordier. Diagnosing discrete event systems : An experiment in telecommunication networks. In *WODES98, Fourth Workshop on Discrete Event Systems*, Cagliari, Italy, 1998.
- [15] M. Sampath, R. Sengupta, S. Lafortune, K. Srinamohideen, and D. Teneketzi. Diagnosability of discrete event system. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.
- [16] M. Sampath, R. Sengupta, S. Lafortune, K. Srinamohideen, and D. Teneketzi. Active diagnosis of discrete-event systems. *IEEE Transactions on Automatic Control*, 43(7):908–929, 1998.