

Coopération homme-machine dans l'analyse diagnostique

Quang-Huy GIAP¹, Stéphane PLOIX², Jean-Marie FLAUS³

G-SCOP, Laboratoire des Sciences pour la Conception,
l'Optimisation et la Production de Grenoble - UMR5272
46, avenue Félix Viallet - 38031 Grenoble Cedex 1 - France

¹Quang-Huy.Giap@g-scop.inpg.fr

²Stephane.Ploix@g-scop.inpg.fr

³Jean-Marie.Flaus@g-scop.inpg.fr

Résumé—Dans la littérature scientifique, il est souvent supposé que les modèles peuvent être établis complètement avant l'analyse diagnostique. Pourtant, dans les problèmes de maintenance industrielle réels, la difficulté à appréhender les systèmes complexes est généralement liée à la formalisation de la connaissance. Si la connaissance du comportement de certains équipements est parfois disponible, il est rare de disposer de modèles de comportement détaillés de tous les équipements. Les étapes de modélisation et d'analyse diagnostique se succèdent souvent car les systèmes sont partiellement représentés et certaines parties sont décrites au fur et à mesure d'un processus de diagnostic. Une analyse diagnostique permettant d'appréhender ces différentes étapes avec le problème de la concomitance de différents niveaux d'abstraction liée à des modélisations partielles, seraient particulièrement adaptées à un grand nombre de besoins réels. Dans cet article, un outil de diagnostic permettant de répondre à ces besoins est présenté.

Index Terms—analyse diagnostique, théorie de l'abstraction, modélisation structurelle, modélisation comportementale, modélisation fonctionnelle.

I. INTRODUCTION

Dans la communauté de diagnostic, l'abstraction a été présentée comme une technique pour réduire le coût de calcul des approches de diagnostics basées sur des modèles [1, 2, 6, 14]. Tout d'abord, la procédure d'abstraction tend à agréger au fur et à mesure des composants pour décrire un système à différents niveaux d'abstraction avec différents niveaux de détails (structurels et comportementaux). Elle est appelée *méthode ascendante* parce qu'elle commence par le niveau de description le plus détaillé des composants et remonte au niveau le plus abstrait. Des algorithmes, développés à partir de l'approche de Mozetic [9], ont été proposés pour résoudre ce problème. Contrairement à la méthode ascendante, une *méthode descendante* est proposée dans cet article. Notre but est d'utiliser l'abstraction pour adapter le problème de diagnostic au contexte de la coopération homme-machine. En effet, il faut remarquer qu'un être humain réalisant une analyse diagnostique commence souvent par des observations globales en tenant en compte d'éléments à un niveau très abstrait. Puis, au fur et à mesure, la description du système se précise dans une direction qui dépend des symptômes observés et des niveaux plus fins de modélisation apparaissent. Le

problème de diagnostic s'inscrit donc dans un processus avec des raffinements consécutifs. Cet article examine comment aider des opérateurs de maintenance dans un tel contexte.

Dans la section 2, les principaux concepts théoriques de l'abstraction sont rappelés pour introduire le problème de diagnostic interactif. Puis, un problème de diagnostic interactif complet est formulé. Un algorithme, tiré de l'algorithme de l'arbre HS-Tree [13], est proposé dans la section 3 pour résoudre le problème. Dans la section 4, un exemple est pris pour illustrer le problème.

II. FORMULATION DU PROBLÈME

Pour modéliser un système physique, plusieurs point de vues sont distingués [3] :

- structurel : que contient le système physique et comment les éléments du système sont connectés ?
- comportemental : comment se comporte chaque élément ?
- fonctionnel : quel est le rôle de chaque élément ?
- téléologique : dans quel but chaque élément a-t-il été installé ?

Dans la littérature, [4] utilise la théorie de l'abstraction pour gérer le modèle comportemental en parallèle avec différents niveaux d'abstraction structurelle dans l'analyse diagnostique [10]. Il permet de réduire la complexité d'un raisonnement. Afin de construire un modèle comportant plusieurs niveaux d'abstraction pouvant être utilisé dans l'analyse diagnostique en se basant sur l'approche DX [13, 5], seulement les trois premiers aspects sont considérés. L'aspect téléologique, qui est plutôt utile pour découvrir les causes des défauts, est rarement étudiée dans l'analyse diagnostique. Dans ce papier, l'utilisation du terme *item* est préférée à *composant*. Un *item* peut décrire des éléments de natures différentes, il peut par exemple représenter une fonction ou une ressource matérielle d'un système physique.

A. Modélisation comportementale et fonctionnelle

Dans un système, la réalité physique est constituée de phénomènes observables, modélisés par des variables. L'ensemble des valeurs possibles qui permettent de décrire un phénomène représenté par la variable v s'écrit $dom(v)$.

Le modèle comportemental est décrit à l'aide de contraintes qui sont représentées par les relations entre les variables. Selon [11], soit $V = \{v_1, v_2, \dots, v_n\}$ est un ensemble de variables, κ est une contrainte sur V si $\kappa \subset \text{dom}(V)$ où $\text{dom}(V) = \text{dom}(V_1) \times \dots \times \text{dom}(V_n)$. L'ensemble V est donc noté par $V = \text{var}(\kappa)$.

Le comportement d'un *item* I est spécifié via des contraintes qui caractérisent des ensembles de valeurs inclus dans les espaces des variables concernées. Ces ensembles seront nommés les *zones*. Parmi les parties de l'espace des variables caractérisant un comportement, seulement certaines parties sont physiquement possibles. Cette zone est appelée zone des possibles ZP . Ces parties peuvent s'exprimer par des relations mathématiques (égalités ou inégalités comportant des expressions calculées en fonction des variables) ou des relations logiques ou qualitatives exprimées sous forme de tables (tables de comportement) [8].

Le complément de la zone des possibles dans l'espace des variables est appelé la zone impossible ZI , exprimé par le fait qu'il est physiquement impossible à atteindre cette zone.

Les modes de comportement d'un *item* désignent une famille signifiante de comportements. Le comportement d'un *item* est défini à l'aide d'un ensemble de contraintes dans l'espace des variables (Figure ??). Dans une zone possible, un *item* peut avoir un mode de fonctionnement/comportement correct et plusieurs modes de fonctionnements incorrects différents. Les modes sont classés en trois catégories :

- Mode de fonctionnement/comportement normal, noté par $ok(\text{item})$ (mode normal) est caractérisé par le fait que les variables restent dans la zone de comportement normal ZN .
- Mode de défaut (ou de défaillance), noté par $fm(\text{Item})$ (fault mode), est caractérisé par le fait que les variables sont dans une zone de comportement anormal bien identifiée.
- Mode de défaut (ou de défaillance) inconnu, noté par ufm (unknown fault mode), est caractérisés par le fait que les variables sont hors zone spécifiée ZN , et tombent dans la zone des possibles ZP mais sans pour autant appartenir à une zone ZF . Les modes ufm sont donc des modes anormaux auxquels ne correspondent aucune contrainte.

Par conséquent, l'ensemble des modes de comportement d'un *item* est écrit par :

$$\text{Modes}(\text{item}_i) = \{ok, [fm_1, \dots, fm_n], cfm\}.$$

Un *item* est appelé non-modélisé s'il n'y a pas de contrainte disponible qui représente ses modes de comportement. Dans le cas le plus courant, il existe 2 modes ok et ufm pour un *item*. Un *item* non-modélisé peut être utilisé pour représenter la partie inconnue dans un système. Cela sera détaillé dans la section suivante.

B. Formulation de l'abstraction

Dans un système, un *item* est généralement une fonction ou une ressource matérielle. La décomposition du système commence souvent par l'identification de la fonction globale du système, puis on décompose en *sous-items*, qui peuvent

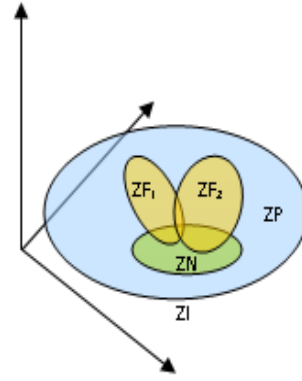


FIGURE 1. Zônes dans l'espace des variables

être des sous-fonctions ou des sous-composants...Le comportement attendu d'un *item* est ainsi réalisé par un ensemble de *sous-items*. La décomposition se termine à un niveau qualifié de terminal. Afin de formaliser les relations hiérarchiques entre les *items*, nous allons introduire la notion de *m-proposition*.

Définition 1: (m-proposition) Une proposition logique où les symboles sont des modes des *items*, qui peut être exprimée par une forme normale conjonctive, est appelée une m-proposition. Si $\mathcal{P}(\text{mode}_1, \dots, \text{mode}_n)$ est une m-proposition, le support de \mathcal{P} est défini par $\text{Modes}(\mathcal{P}) = \{\text{mode}_1, \dots, \text{mode}_n\}$.

Par exemple, $(\text{mode}_1 \rightarrow \text{mode}_2) \wedge \text{mode}_3$ avec $\neg \text{mode}_1 = \text{mode}_4 \vee \text{mode}_5$ est une m-proposition, parce qu'elle peut être réécrite par : $(\text{mode}_4 \vee \text{mode}_5 \vee \text{mode}_2) \wedge \text{mode}_3$.

Définition 2: (monôme de m-proposition) Un monôme dans une m-proposition est l'une des propositions disjonctives apparaissant dans la forme normale conjonctive équivalente.

Par l'exemple précédent, $\text{mode}_4 \vee \text{mode}_5 \vee \text{mode}_2$ et mode_3 sont les monômes de la m-proposition.

Le concept de l'abstraction comportementale partielle peut être introduit ensuite.

Définition 3: (abstraction comportementale partielle) Soient I un *item* et $\mathbb{I} = \{I_1, \dots, I_n\}$ un ensemble d'*items*. I est une abstraction comportementale partielle de \mathbb{I} si $\forall m_i \in \text{Modes}(I)$ et il existe un \mathcal{P}_i tel que : $m_i \rightarrow \mathcal{P}$ avec $\text{Modes}(\mathcal{P}_i) = \{\text{mode}(I_1), \dots, \text{mode}(I_n)\}$.

Si I est une abstraction comportementale partielle de $\mathbb{I} = \{I_1, \dots, I_n\}$, I est appelée *item-parent* de chaque *item* I_i . À l'inverse, chaque *item* I_i est appelé *item-enfant* de I . Normalement, si un *item-parent* se comporte correctement, on en déduit que ses *item-enfant* sont en mode de fonctionnement normal. Il est représenté par une implication logique $ok(I) \rightarrow ok(I_1) \wedge ok(I_2) \wedge \dots \wedge ok(I_n)$. Dans le contexte d'une coopération homme-machine, l'abstraction comportementale partielle représente la connaissance de l'expert, qui teste le système en défaut, sur la structure d'un système.

Définition 4: (abstraction comportementale complète) Soient I un *item* et $\mathbb{I} = \{I_1, \dots, I_n\}$ est un ensemble d'*items*.

I est une abstraction comportementale complète de \mathbb{I} si $\forall m_i \in Modes(I)$ et il existe une m-proposition \mathcal{P}_i telle que $m_i \leftrightarrow \mathcal{P}$ avec $Modes(\mathcal{P}_i) = \{mode(I_1), \dots, mode(I_n)\}$.

Une abstraction comportementale partielle $\mathbb{I} = \{I_1, \dots, I_n\}$ de I peut toujours être transformée en une abstraction comportementale complète en introduisant un *item* virtuel qui représente la partie de I , qui n'est pas incluse dans \mathbb{I} , notée par VI pour *item virtuelle*, avec $VI = I \setminus \mathbb{I}$.

C. Propagation des défauts

Dans les systèmes physiques réels, la propagation des défauts est représentée par le fait que le mode de défaut d'un *item* va impliquer le mode de défaut d'un autre *item*. La propagation des défauts est généralement représentée par une implication logique, par exemple $mode(item_i) \rightarrow mode'(item_j)$. Pour tenir compte du phénomène de propagation des défauts, la transformation des implications logiques en conjonctions logiques est proposée. Une implication logique $A \rightarrow B$ est équivalente à $\neg A \vee B$, donc $mode(item_i) \rightarrow mode'(item_j)$ est équivalent à $\neg mode(item_i) \vee mode'(item_j)$.

D. Formulation d'un problème de diagnostic complet

Résumons maintenant les éléments d'un problème de diagnostic :

- 1) la liste des *items* et des modes possibles pour chaque *item*.
- 2) les abstractions comportementales partielles déduites de la connaissance de l'expert.
- 3) les modes impliqués dans les tests inconsistants, modélisés par les m-propositions.
- 4) les propagations des défauts modélisées par les m-propositions.

III. PROCESSUS DE DIAGNOSTIC INTERACTIF

Dans cette partie, nous allons détailler les processus de diagnostic basés sur la décomposition interactive (méthode descendante). Il s'agit d'un processus interactif entre un outil de diagnostic (une machine) et un expert. Le processus commence lorsque un dysfonctionnement est détecté. La tâche d'isolation des défauts commence généralement par des tests qui vérifient la fonction globale du système. A chaque interaction de l'expert, il effectue des tests, collecte des nouveaux symptômes et continue le processus. Selon le principe de monotonie, l'outil de diagnostic fournit les diagnostics qui sont de plus en plus détaillés. Étape par étape, l'expert localise les sous-systèmes ou composants qui sont en mode de défaut. Ce processus de diagnostic est décrit par la figure 2.

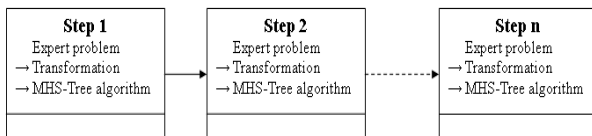


FIGURE 2. Processus de diagnostic

La résolution de problèmes est la même à chaque interaction dans le processus. Concentrons-nous maintenant sur ce qui se passe entre deux interactions. Le processus de diagnostic entre deux interactions peut être décomposé en deux étapes. La première étape est appelée *transformation* : il transforme le problème énoncé de l'expert avec les abstractions comportementales partielles en un problème résoluble. La deuxième étape est basée sur un *algorithme de MHS-Tree*, qui calcule et fournit des diagnostics à partir du problème résoluble transformé comme précédemment.

A. Transformation

Au cours de l'étape de transformation, la connaissance initiale de l'expert sur le système (symptômes, décomposition du modèle et la propagation des défauts, etc..) peut être transformée en un m-proposition en :

- 1) introduisant le mode de défaut complémentaire pour chaque *item* connu : il existe au moins deux modes *ok* et *cfm* pour un *item*. En conséquence, si les modes de comportement ne sont pas définis pour un *item*, il sera automatiquement ajouté à cette transformation. Considérons par exemple la décomposition d'une voiture dont les modes déclarés par un expert sont : $modes(car) = \{ok, no-start\}$ (cet exemple sera exploité pour illustrer les tâches de l'étape de transformation). Supposons que la voiture est décomposée en un système d'allumage (*IS*) et un système de démarrage (*ESS*). La connaissance initiale de l'expert sur les modes de comportement sont déclarés par $modes(IS) = \{ok, no-spark\}$, $modes(ESS) = \{ok, no-crank\}$. L'étape de transformation va ajouter automatiquement les modes *ok* et *cfm* pour le système d'allumage et le système de démarrage. Il vient donc : $modes(car) = \{ok, no-start, cfm\}$, $modes(IS) = \{ok, no-spark, cfm\}$, $modes(ESS) = \{ok, no-crank, cfm\}$.

Pour cela, on a :

- 2) introduit un *item* virtuel complémentaire pour transformer les abstractions partielles en abstractions complète. Toutes les implications de la conjonction des modes enfant à chaque mode de parent ont été formalisées afin de calculer les m-propositions équivalentes.
- 3) transformé les implications logiques de la propagation de défauts en propositions disjonctives (voir ??).
- 4) remplacé des modes abstraits par leurs m-propositions équivalentes pour les points (3) et (4) de la section II-D.
- 5) développé les m-propositions en une forme normale conjonctive et découpé la proposition obtenue en un ensemble de monômes.

Enfin, après ces transformations, le problème de diagnostic à résoudre peut être formulé sous forme d'une m-proposition dont les monômes sont fournis à l'algorithme HS-Tree pour calculer les diagnostics.

B. Algorithme de résolution

Dans les problèmes où plusieurs modes peuvent s'associer à un *item*, l'algorithme standard de HS-Tree (un arbre dont

les noeuds sont les *hitting-sets* [13]), peut conduire à des diagnostics qui contiennent en même temps plusieurs modes de comportement d'un même *item*. Ces diagnostics sont impossibles car un *item* ne peut être qu'en un seul mode à un instant donné. La prise en compte du modèle multi-modes dans l'analyse diagnostique n'est pas un problème nouveau. Dans la littérature, certaines approches de résolution ont été proposées dans [7, 14]. En se basant sur ATMS [6], le modèle multi-mode a été intégré dans GED+ [14] pour analyser si les défauts des composants ont vraiment expliqué les observations. L'algorithme Sherlock [7] est développé à partir de GDE pour calculer automatiquement les ensembles de conflits et des hypothèses de diagnostic. Il est basé sur le raisonnement des probabilités pour limiter la combinatoire. Dans cette section, une adaptation de l'algorithme de HS-Tree [13] a été préférée pour sa clarté pour gérer les défauts multiples. Le chemin d'un noeud à la racine d'un arbre HS montre clairement tous les éléments impliqués dans les diagnostics temporaires obtenus dans la construction d'un arbre HS. Ensuite, il est facile d'éviter l'existence de deux ou plusieurs modes d'un *item* dans le résultat de diagnostic. En outre, en comparaison avec l'algorithme original de HS-Tree, qui est basé sur un ensemble de conflits, MHS-Tree est étendu à un ensemble de propositions disjonctives pour calculer les *hitting-sets*. Chaque proposition disjonctive peut correspondre à un test inconsistant, ou à une proposition transformée des propagations des défauts.

Afin de garantir un raisonnement rigoureux, un test consistant ne sera pas utilisé pour calculer des diagnostics sauf s'il est entièrement testé. Pourtant, les résultats des tests normaux consistants sont utiles pour la classification des diagnostics. Dans [12], une approche qui est basée sur la distance entre les signatures théoriques et effectives a été proposée. Elle est étendue au contexte multi-mode.

Soit $T = (t_i)$ est une liste ordonnée des tests, et $M = (m_i)$ un ensemble de modes de défaut. La signature de M dans T est donnée par $\sigma_T(M)$:

$$\forall i, \begin{cases} (\sigma_T(M))_i = 1 \leftrightarrow M \cap \overline{\prod_{mode}(t_i)} \neq \emptyset \\ (\sigma_T(M))_i = 0 \leftrightarrow M \cap \prod_{mode}(t_i) = \emptyset \end{cases} \quad (1)$$

où $\prod_{mode}(t_i)$ correspond à l'ensemble des modes impliqués dans le test t_i . $\overline{\prod_{mode}(t_i)}$ correspond à l'union des modes complémentaires de chaque mode impliqué dans le test t_i :

$$\overline{\prod_{mode}(t_i)} = \bigcup_{m(I) \in \prod_{mode}(t_i)} Modes(I) \setminus \{m(I)\}$$

Soit $T = (t_i)$ une liste ordonnée des tests. A un moment donné, la signature effective dans T , notée σ_T^* , est donnée par :

$$\forall i, \begin{cases} (\sigma_T^*)_i = 1 \leftrightarrow t_i \text{ is inconsistent} \\ (\sigma_T^*)_i = 0 \leftrightarrow t_i \text{ is consistent} \end{cases} \quad (2)$$

La prochaine mesure tente de mesurer la ressemblance entre la signature théorique et la signature effective d'un diagnostic [12]. Soit $T = (t_i)$ une liste ordonnée de tests, et $D = d_i$ est un

ensemble de diagnostics. La mesure de coïncidence est donnée par :

$$\forall d_i \in D, \mu_T^c(d_i) = \frac{|\sigma_T(d_i), \sigma_T^*|_{Hamming}}{dim(T)} \quad (3)$$

Cette mesure sera illustrée dans l'exemple suivant.

IV. APPLICATION

Afin d'illustrer comment l'approche proposée convient à l'analyse diagnostique itérative avec des décompositions consécutives, prenons l'exemple d'une voiture en défaut étudiée par un garagiste.

Tout d'abord, l'opérateur note que la voiture ne démarre pas. À cette étape, le symptôme obtenu, qui est aussi un diagnostic trivial, est *ufm(voiture)*. Il est très général et ne conduit pas directement à l'étape suivante : presque tous les défaut sont possible. Implicitement, les modes possibles de la voiture sont les suivants :

$$Modes(car) = \{ok, cfm\} \quad (4)$$

Deuxièmement, parce que le système de démarrage peut être testé plus facilement, l'expert décompose implicitement la *voiture* en : alimentation électrique (*EPR*), système de démarrage à l'exception du démarreur (*ESS*) et démarreur (*SD*). La décomposition peut être modélisée par :

$$ok(car) \rightarrow ok(EPR) \wedge ok(ESS) \wedge ok(SD) \quad (5)$$

Ensuite, l'expert tourne la clé pour tester si le démarreur fonctionne. Cela correspond à un nouveau test. Comme il entend le bruit du système de démarrage, il déduit à partir de *test 1* que :

$$\exists OBS/ok(EPR) \wedge ok(ESS) \wedge ok(SD) \quad (6)$$

Le test positif (consistant) peut être utilisé pour classer les diagnostics en utilisant la mesure de coïncidence. Les symptômes observés sont :

$$cfm(car) \quad (7)$$

$$\exists OBS/ok(EPR) \wedge ok(ESS) \wedge ok(SD) \quad (8)$$

L'expression (8) signifie qu'il existe au moins une observation telle que le test donné par (6) est consistant.

Le problème est maintenant défini par (4), (5), (6), (7) et (8). Nous allons alors transformer ce problème en un problème résoluble. Afin d'obtenir une abstraction comportementale complète, des modes de défaut complémentaires et un *item virtuel* sont introduits. Il est nommé $VI_1 = car \setminus \{EPR, ESS, SD\}$. Le nouvel ensemble de modes transformés déduit de (4) est :

$$\begin{aligned} \{Modes(EPR) = (ok, cfm); Modes(ESS) = (ok, cfm); \\ Modes(SD) = (ok, cfm); Modes(VI_1) = (ok, cfm)\} \end{aligned} \quad (9)$$

La décomposition peut donc être réécrite avec des équivalences :

TABLE I
SIGNATURE TABLE 1

	ok(EPR)	ok(ESS)	ok(SD)	ok(VI ₁)
T ₁	1	1	1	0

$$ok(car) \leftrightarrow ok(EPR) \wedge ok(ESS) \wedge ok(SD) \wedge ok(VI_1) \quad (10)$$

$$cfm(car) \leftrightarrow cfm(EPR) \vee cfm(ESS) \vee cfm(SD) \vee cfm(VI_1) \quad (11)$$

En utilisant l'algorithme de MHS-arbre, les diagnostics peuvent être calculés à partir du problème transformé. Cela conduit à :

$$\{cfm(EPR)\}; \{cfm(ESS)\}; \{cfm(SD)\}; \{cfm(VI_1)\} \quad (12)$$

Pour classer les diagnostics, une table de signature (I) peut être obtenue à partir de (6), (7) et (8) :

La signature théorique des défauts est :

$$\sigma_T(cfm(EPR)) = (1);$$

$$\sigma_T(cfm(ESS)) = (1);$$

$$\sigma_T(cfm(SD)) = (1);$$

$$\sigma_T(cfm(VI_1)) = (0).$$

Comme le test 1 est consistant, la signature effective est $\sigma_T^* = (0)$.

De (3), les mesures de coïncidence sont données par :

$$\mu_T^c(cfm(EPR)) = 1.00;$$

$$\mu_T^c(cfm(ESS)) = 1.00;$$

$$\mu_T^c(cfm(SD)) = 1.00;$$

$$\mu_T^c(cfm(VI_1)) = 0.00.$$

Comme $\mu_T^c(cfm(VI_1)) = 0.00$ est la valeur la moins importante, l'expert décide à tester les sous-parties de l'*item virtuel*, il s'agit de la partie de la voiture qui n'est pas *EPR*, *ESS* ou *SD*. Supposons que l'opérateur se concentre sur le système d'allumage. L'opérateur démonte la bougie d'allumage, et tient le bout de la bougie d'allumage avec son câble proche d'une surface métallique et démarre le véhicule sans utiliser le système de démarrage électrique. L'opérateur ne voit aucune étincelle venant des bougies d'allumage. Ces tests sont négatifs. Il en déduit que la source d'énergie électrique (*EPR*), le circuit d'allumage (*IC*) ou les bougies d'allumage (*SP*) sont en défaut. Ainsi, l'*item virtuel* est décomposé en le circuit d'allumage (*IC*) et les bougies d'allumage (*SP*) :

$$ok(VI_1) \rightarrow ok(SP) \wedge ok(IC) \quad (13)$$

Le nouveau test conduit à :

$$\neg ok(EPR) \vee \neg ok(SP) \vee \neg ok(IC) \quad (14)$$

Par conséquence, un nouveau ensemble de symptômes est donné par : (7), (8) et

$$cfm(VI_1) \quad (15)$$

$$\neg ok(EPR) \vee \neg ok(SP) \vee \neg ok(IC) \quad (16)$$

TABLE II
SIGNATURE TABLE 2

	ok(EPR)	ok(ESS)	ok(SD)	ok(SP)	ok(IC)	ok(VI ₂)
T ₁	1	1	1	0	0	0
T ₂	1	0	0	1	1	0

Le nouveau problème à résoudre est donné par : (4), (5), (6), (7), (8), (13), (14), (15), et (16).

Ce problème est transformé par l'ajout d'un *item virtuel* $VI_2 = VI_1 \setminus \{SP, IC\}$, qui est égal à $car \setminus \{EPR, ESS, SD, SP, IC\}$.

Après la transformation, le nouvel ensemble de modes est donné par (4), (9) et :

$$\{ (SP) = (ok, cfm); (IC) = (ok, cfm); (VI_2) = (ok, cfm) \} \quad (17)$$

L'abstraction transformée est donnée par (10), (11) et

$$ok(VI_1) \leftrightarrow ok(SP) \wedge ok(IC) \wedge ok(VI_2) \quad (18)$$

$$cfm(VI_1) \leftrightarrow cfm(SP) \vee cfm(IC) \vee cfm(VI_2) \quad (19)$$

En utilisant l'algorithme de MHS-tree, les diagnostics sont calculés à partir du problème transformé :

$$\{cfm(EPR)\}; \{cfm(SP)\}; \{cfm(IC)\} \quad (20)$$

De (6), (7) et (8), une table de signatures est obtenue :

$$(1 \ 1)$$

Les signatures théoriques des défauts sont :

$$\sigma_T(cfm(EPR)) = (1 \ 1);$$

$$\sigma_T(cfm(ESS)) = (1 \ 0);$$

$$\sigma_T(cfm(SD)) = (1 \ 0);$$

$$\sigma_T(cfm(SP)) = (0 \ 1);$$

$$\sigma_T(cfm(IC)) = (0 \ 1);$$

$$\sigma_T(cfm(VI_2)) = (0 \ 0).$$

La signature effective est : $\sigma_T^* = (0 \ 1)$.

Les mesures de coïncidence sont données par :

$$\mu_T^c(cfm(EPR)) = 0.50;$$

$$\mu_T^c(cfm(ESS)) = 1.00;$$

$$\mu_T^c(cfm(SD)) = 1.00;$$

$$\mu_T^c(cfm(SP)) = 0.00;$$

$$\mu_T^c(cfm(IC)) = 0.00;$$

$$\mu_T^c(cfm(VI_2)) = 0.50.$$

Comme $\mu_T^c(cfm(SP)) = 0.00$, $\mu_T^c(cfm(IC)) = 0.00$ sont les valeurs les plus faibles à la fin de cette étape, la partie qui est en défaut correspond le plus vraisemblablement au circuit d'allumage (*IC*) ou aux bougies d'allumage (*SP*).

V. CONCLUSION

L'approche proposée permet d'accompagner un opérateur tout au long d'un processus de diagnostic dans un contexte de coopération homme-machine et de résoudre le problème de diagnostic sans avoir le modèle complet du système dès le début. Un processus itératif a été proposé pour traiter

l'information étape par étape grâce à des décompositions hiérarchiques. Les diagnostics sont ainsi affinés au fur et à mesure. Des problèmes de diagnostic déduits de l'expertise formalisée sont fournis à chaque itération. Ils sont résolus par des transformations en problèmes résoluble composés des connaissances disponibles (décomposition, tests inconsistants, tests consistants, propagation des défauts, etc. . .). Le problème transformé peut alors être résolu par l'algorithme de MHS-Tree proposé. Le processus de diagnostic itératif a été illustré par un exemple.

RÉFÉRENCES

- [1] K. Autio. Abstraction of behaviour and structure in model-based diagnosis. In *the Sixth International Workshop on Principle of Diagnosis (DX95)*, pages 1–7, 1995.
- [2] K. Autio and R. Reiter. Structural abstraction in model based diagnosis. In *The 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 269–273. John Wiley and Sons, 1998.
- [3] L. Chittaro and A.N. Kumar. Reasoning about function and its applications to engineering. *Artificial Intelligence in Engineering*, 12(4) :331–336, 1998.
- [4] L. Chittaro and R. Ranon. Hierarchical model-based diagnosis based on structural abstraction. *Artificial Intelligence*, 1-2 :147–182, 2004.
- [5] J. De Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32 :97–130, 1987.
- [6] Johan de Kleer. Problem solving with the atms. *Artif. Intell.*, 28(2) :197–224, 1986.
- [7] Johan de Kleer and Brian C. Williams. Diagnosis with behavioral modes. *Readings in model-based diagnosis*, pages 124–130, 1992.
- [8] M. Desindes. *Contribution à la mise au point d'une approche intégrée analyse diagnostique / analyse de risques*. PhD thesis, Grenoble Institute of Technology (INPG), 2006.
- [9] I. Mozetic. *Hierarchical model-based diagnosis*, pages 354–372. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.
- [10] P. P. Nayak and A. Y. Levy. A semantic theory of abstractions. In *14th International Joint Conference on Artificial Intelligence IJCAI-95*, pages 196–203, Montreal, Canada, 1995.
- [11] S. Ploix and P. Chazot. Iterative expert driven fault diagnosis based on structural modeling. In *IFAC Symposium SAFEPROCESS'2006*, August 2006.
- [12] S. Ploix, S. Touaf, and J. M. Flaus. A logical framework for isolation in fault diagnosis. In *SAFEPROCESS'2003*, Washington D.C., U.S.A., 2003.
- [13] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32 :57–95, 1987.
- [14] P. Struss and O. Dressler. “physical negation” : Integrating fault models into the general diagnostic engine. In *Proc. of the 11th IJCAI*, pages 1318–1323, Detroit, MI, 1989.