Caractérisation des systèmes autoguérissants : diagnostiquer ce que l'on peut réparer

Marie-Odile Cordier[⋆], Yannick Pencolé[†], Louise Travé-Massuyès[†], Thierry Vidal[⋆]

[⋆] Université de Rennes 1/IRISA
Campus de Beaulieu F-35042 Rennes cedex
marie-odile.cordier@irisa.fr
[†] IRISA/INRIA
Campus de Beaulieu F-35042 Rennes cedex
thierry.vidal@irisa.fr
[‡] LAAS-CNRS, Université de Toulouse
7 avenue du Colonel Roche F-31077 Toulouse Cedex 4
louise.trave-massuyes@laas.fr et
yannick.pencole@laas.fr

Résumé

Les architectures informatiques nécessitent aujourd'hui des capacités "d'autoguérison", c'est-à-dire de diagnostic de l'occurrence de fautes et de réparation de leurs effets, de manière autonome, pour continuer à assurer leurs fonctionnalités. Cela est particulièrement vrai dans le domaine des Services Web, auxquels nous appliquons actuellement ces travaux. Les concepteurs de tels systèmes ont besoin d'outils permettant de vérifier avant leur mise en œuvre opérationnelle qu'ils sont bien "autoguérissants". Pour cela, une première étape consiste à définir formellement ce qu'autoguérison signifie. La diagnosticabilité est la capacité d'un système à déterminer l'état fautif dans lequel il se trouve à partir des observations dont il dispose. La réparabilité est la capacité d'un système à disposer de plans de réparation adaptés aux fautes. Nous proposons une nouvelle définition de diagnosticabilité, qui ne s'appuie plus sur une partition des fautes comme le fait la définition classique, mais sur un ensemble couvrant de macrofautes et permet d'associer à un système son niveau de diagnosticabilité. Nous proposons ensuite une première définition formelle de la réparabilité. Ces deux définitions sont celles qui conviennent pour caractériser la capacité d'autoguérison d'un système. Nous proposons ainsi une définition d'autoguérison en combinant pour la première fois explicitement la diagnosticabilité et la réparabilité. Un théorème peut alors être démontré dont découle assez directement un algorithme de vérification de l'autoguérison. Nous terminons en montrant comment ce travail peut servir à élaborer des stratégies et aider les concepteurs à analyser leur système et le rendre autoguérissant.

Mots-clés *: Diagnostic, Réparation, Autoguérison, Agents autonomes, Systèmes à événements discrets*

Abstract

Nowadays, computer architectures need to be self-healing, which means capable of surviving autonomously the occurrence of faults, still managing to provide the desired functionalities. That is true for instance for Web Services, to which we are currently applying this theoretical work. Designers of such systems need tools helping them to assess beforehands the self-healability of their system. A first step towards such tools is to formally define what self-healability precisely means. Diagnosability is the ability of a system to be self-aware about its current state by analysing the observations that are received ; repairability can be defined as the ability of a system to react to faults by applying repair actions. We define self-healability as a joint property, which achieves for the first time a bridge between diagnosability, for which we had to extend the classical definition, and repairability, for which we provide a first formal definition. A theorem can then be proved, leading naturally to a practical and tractable self-healability checking algorithm. Finally, it is shown how the above results can be used to suggest strategies to help designers building self-healing systems.

Key-words*: Diagnosis, Repair, Self-healability, Autonomous agents, Discrete event systems*

1 INTRODUCTION

Les systèmes dynamiques complexes, de plus en plus présents au sein des applications industrielles, se doivent de disposer d'un niveau élevé d'autonomie, y compris en présence d'états fautifs avérés. Ils doivent pouvoir connaître à tout instant leur état courant et être réactifs vis-à-vis des fautes, par l'application de plans de réparation susceptibles de les ramener à leur fonctionnement nominal. En d'autres termes, ces systèmes doivent être "autoguérissants"[4] [7].

De nombreuses applications illustrent ce besoin, de la robotique mobile à la surveillance de centrales énergétiques ; nous nous sommes plus particulièrement intéressés au développement des services web (notamment dans le cadre du projet européen WS-DIAMOND, voir [18] pour plus de détails), comme par exemple la réservation de voyages sur Internet : un ensemble de services distribués se coordonnent par l'intermédiaire de processus décrits à l'aide de diagrammes d'activités (*workflows*) prédéfinis, pour fournir à un utilisateur une réponse à sa requête. Qui plus est, des exigences en termes de qualité de service doivent généralement être satisfaites (par exemple le délai de réponse, ou le nombre de solutions testées). De tels systèmes reposent

[4] Traduction littérale du terme anglais *self-healing*.

clairement sur une approche de type “récupération” des fautes (par opposition à la tolérance aux fautes) : lorsqu'une faute apparaît (par exemple une donnée est corrompue lors d'une communication, ou un service est temporairement inaccessible), une exception est émise et un plan de réparation (précompilé ou bien construit dynamiquement) est exécuté, censé ramener le système dans un état de fonctionnement normal. Ces plans utilisent des actions de base, du style *refaire l'activité “Louer voiture” avec comme nouveau paramètre “économique”*, ou *remplacer le service “Réservation de train” par le service “Réservation de bus”*. Il est crucial d'être capable de vérifier, dès la conception, que de tels services web sont sûrs, c'est-à-dire qu'ils sont autoguérissants. Pour cela, il convient d'analyser et définir précisément le concept d'autoguérison dans les systèmes orientés récupération, de manière générique et formelle.

Concevoir des systèmes autoguérissants exige d'évaluer conjointement les deux propriétés connues comme la diagnosticabilité et la réparabilité. La première définit la capacité d'un système à produire des “observables” distincts pour chaque situation fautive envisagée ; la seconde définit la capacité à sortir de telles situations fautives en choisissant parmi un ensemble d'actions correctrices connues. Pour autant ces deux propriétés sont généralement étudiées de manière indépendante. La communauté du diagnostic a pour sa part proposé une définition formelle de la diagnosticabilité pour les systèmes événementiels [15] et pour les systèmes basés états [20] que [5] a démontré être conceptuellement équivalentes. Elle a également développé différentes approches pour sa vérification. La réparabilité par contre a été beaucoup moins étudiée, les chercheurs du domaine du génie logiciel se tournant plus volontiers vers des approches de type contrôlabilité ou tolérance aux fautes, alors que le domaine de la planification a su modéliser des plans de réparation mais sans s'intéresser explicitement aux conditions formelles de leur applicabilité.

Une première approche naïve consiste, après avoir identifié l'ensemble de toutes les fautes possibles, à analyser séparément leur diagnosticabilité (les observables issus de fautes peuvent-ils toujours être rattachés à ces fautes sans ambiguïté ?), puis leur réparabilité (existe-t-il pour chaque faute, après chacune de ses occurrences possibles, au moins un plan d'actions capable de la réparer ?). Si les deux réponses sont positives, alors le système est autoguérissant. Mais il s'agit d'une exigence trop forte : la diagnosticabilité de chaque faute élémentaire n'est en effet pas nécessairement requise... Imaginons par exemple que nos observations permettent de toujours déterminer que l'une ou l'autre des fautes f_1 et f_2 a eu lieu, mais sans pour autant pouvoir les discriminer. Notre système n'est alors pas diagnosticable vis-à-vis des fautes élémentaires. Pourtant, si le plan de réparation r s'avère adapté pour réparer aussi bien f_1 que f_2, alors cela suffit pour assurer la capacité d'autoguérison ! Cet exemple illustre la nécessité de définitions conjointes de la diagnosticabilité, de la réparabilité, et de l'autoguérison. L'idée sous-jacente que nous allons exploiter dans cet article est d'identifier un ensemble de ce que nous appellerons des *macrofautes* (comme “f_1 ou f_2”), tel que ces

macrofautes soient à la fois diagnosticables et réparables.

Notre travail montre qu'il est nécessaire tout d'abord de redéfinir le concept classique de diagnosticabilité : ce dernier est défini en effet à partir de partitions des fautes élémentaires, alors que nous montrons qu'il est intéressant de le définir à partir d'ensembles de macrofautes susceptibles de se recouvrir, deux macrofautes pouvant contenir des fautes élémentaires communes. Il est alors possible d'associer à un système son niveau de diagnosticabilité. Pour évaluer la capacité d'autoguérison, on vérifie qu'il existe au moins un ensemble de macrofautes qui est diagnosticable et peut être réparé. Cela étant, des formes affaiblies d'autoguérison peuvent également être proposées. La première, dénommée autoguérison *faible*, exige la réparabilité forte de l'ensemble des macrofautes mais autorise que ces dernières ne soient diagnosticables que pour un sous-ensemble des comportements possibles du système. La seconde propriété que nous appelons autoguérison *partielle* garantit la diagnosticabilité et la réparabilité des macrofautes en toutes occasions mais pour un ensemble de macrofautes ne couvrant qu'un ensemble réduit de fautes élémentaires.

La section 2 commence par situer nos travaux dans le domaine, puis la section 3 pose un certain nombre de définitions préliminaires et d'hypothèses concernant les fautes, les observables et les actions correctrices. La section 4 est le cœur de l'article. Une nouvelle définition de diagnosticabilité est proposée, qui étend la définition classique. La notion de niveau de diagnosticabilité d'un système est introduite, en s'appuyant sur le treillis des partitions de signatures élémentaires des fautes. Une définition de la réparabilité est donnée, ce qui permet ensuite de définir la notion d'autoguérison. Enfin, un théorème est démontré qui exhibe un ensemble diagnosticable dont il suffit de déterminer la réparabilité pour assurer celle du système. Il en découle un algorithme, décrit dans la section 5, permettant de vérifier la capacité d'autoguérison en temps polynômial. Lorsqu'un système apparaît ne pas être autoguérissant, la section 6 propose d'abord de définir formellement les propriétés d'autoguérison faible et partielle, avant d'aborder de manière plus opérationnelle dans la section 7 les stratégies possibles pour aider un concepteur à identifier les causes et à trouver une solution adéquate pour rendre le système autoguérissant, soit en rajoutant des moniteurs pour disposer de plus d'observations, soit en diversifiant les plans de réparation à sa disposition. Une conclusion est donnée dans la section 8 à travers des perspectives d'extension et d'application concrète de notre travail.

2 ÉTAT DE L'ART

Dans [7], les systèmes autoguérissants sont présentés comme un nouveau domaine de recherche en informatique, leur définition restant relativement générale, à savoir :

Un système est autoguérissant s'il a la capacité de percevoir ses propres dysfonctionnements et, sans intervention humaine, d'effectuer les ajustements nécessaires pour recouvrer son fonctionnement normal.

Cette définition est liée à la notion plus générale encore de *fiabilité*. Un système est fiable (ou sûr) si sa capacité à délivrer son service en permanence est garantie. L'autoguérison est aussi liée à la notion de *tolérance aux fautes* (ou encore *tolérance aux pannes*). Un système est dit tolérant aux fautes s'il demeure fonctionnel malgré certaines fautes de ses constituants.

Il existe différentes façons de garantir ce type de propriétés. L'approche *active* pour améliorer la tolérance aux fautes est d'utiliser pendant la phase opérationnelle des mécanismes, prédéfinis ou élaborés en ligne, se déclenchant après l'occurrence d'une faute. La *contrôlabilité* des systèmes dynamiques proposée dans [12] est un moyen de modifier de façon proactive l'architecture d'un système lors de sa phase de spécification afin d'empêcher l'occurrence d'une faute irréversible.

Par ailleurs, les approches dites *passives* s'appuient sur des mécanismes de contrôle *robustes* qui prennent en compte la possibilité de fautes et garantissent les performances requises en situation normale comme dans de telles situations (voir par exemple [6] dans le domaine des systèmes continus). En génie logiciel, [17] s'appuie sur la même idée en argumentant que la distinction entre un état de bonne et de mauvaise santé est parfois délicate et que l'objectif est plutôt de maintenir un état interne stable du système malgré les variations extérieures (principe de l'homéostasie[5]).

Dans les systèmes autoguérissants et contrairement à ce qui précède, les aspects de remise en état sont clairement mis en avant : la distinction entre les états de bonne et de mauvaise santé est nette. Un tel système dispose de moyens de surveillance afin de (1) détecter le passage d'un mode nominal à un mode de faute, (2) diagnostiquer la situation et (3) choisir et exécuter une stratégie de réparation appropriée.

Ce principe peut être considéré comme une *approche active*. Cette approche peut néanmoins conduire, comme il est souligné dans [13], à l'intégration de modes de réparation dans le modèle global du système, ce qui rend les limites entre l'approche active et passive plus ou moins floues.

Notre contribution est clairement positionnée sur l'autoguérison par l'adoption du point de vue *diagnostic/réparation*. Le modèle spécifiant les comportements du système (de faute ou non) est clairement séparé des stratégies disponibles pour la remise en état nominal du système à partir d'un état de faute. Dans la communauté de la planification, la *révision de plans*, l'*adaptation de plan* ou encore les *techniques de replanification* sont des variations de la même idée : en cas de perturbation dans l'exécution courante d'un plan, le plan nominal est suspendu et une séquence alternative d'actions est exécutée, cette dernière étant soit précalculée (hors ligne) soit calculée en ligne. Dans

[5] Traduction littérale de *homeostasis*.

cette optique, notre contribution est à rapprocher de travaux dans lesquels des *plans alternatifs* ont été précalculés et intégrés dans l'espace d'états du système [21]. Un module de surveillance est en charge de détecter les déviations du système et fait commuter vers un des plans alternatifs si nécessaire afin de remettre le système dans un état dans lequel le plan nominal peut redémarrer. Néanmoins ce type d'approche a généralement deux défauts :

- l'observabilité du système est supposée totale rendant ainsi l'activité de diagnostic triviale ;
- il n'y a pas d'analyse formelle d'une propriété qui garantirait qu'il existe toujours une exécution de plan, réparant avec succès le système, quelle que soit la faute.

Cet article traite de ces deux points. La suppression de l'hypothèse d'observabilité totale implique une analyse de la *diagnosticabilité* du système : cette notion couvre un ensemble de propriétés qui sont étudiées depuis des années par les différentes communautés du diagnostic. Dans le cadre des systèmes continus, la diagnosticabilité est définie par la capacité de détection et d'isolation de fautes [1, 2]. Dans le cadre des systèmes à événements discrets (SED), les premières définitions ont été proposées par [15] et de nombreuses extensions ont suivi notamment celle pour les fautes intermittentes [4] ou encore celle pour aborder des motifs plus génériques [8]. Une définition de la diagnosticabilité unifiant celles des systèmes continus et des systèmes à événements discrets est décrite dans [5]. La vérification de la diagnosticabilité est un problème complexe et plusieurs algorithmes existent [3, 9, 16, 11, 23], l'un des objectifs de cette vérification étant de reboucler sur la conception du système, par l'ajout de capteurs [19, 20], ou encore par la respécification de protocoles de communications entre composants du système [10].

Pour le second point, la garantie du succès d'une réparation requiert une définition formelle de la notion de *réparabilité* qui, à notre connaissance, n'a jamais été clairement établie.

Finalement, notre objectif est de combiner la diagnosticabilité et la réparabilité afin d'établir les conditions nécessaires à l'autoguérison afin d'en extraire des recommandations pour la spécification de systèmes autoguérissants.

3 PRÉLIMINAIRES

Le cadre formel introduit dans cet article peut être utilisé dans le cas de systèmes à dynamique continue ou discrète (en adoptant le point de vue décrit dans [5]). Néanmoins, l'un de nos objectifs est d'appliquer ce cadre à des architectures orientées services (SOA pour *Service Oriented Architectures*) telles que des Services Web pour lesquelles des techniques de diagnostic et réparation sont étudiées et développées au sein du projet européen WS-DIAMOND [18]. Les modèles opérationnels sémantiques des SOAs reposent le plus souvent sur des *workflows*, c'est-à-dire des ensembles d'ac-

tivités atomiques qui s'agencent en fonction de leurs échanges de données en entrée et en sortie [14, 22]. Ces modèles peuvent être décrits sous la forme de systèmes à événements discrets, tels que des automates à états finis ou des réseaux de Petri. C'est pourquoi nous allons nous restreindre à une illustration de nos concepts par le biais de systèmes à événements discrets.

3.1 Événements : Observations et Fautes

L'approche de type système à événements discrets considère que le système évolue en réagissant à l'occurrence d'événements. Dans le modèle à base d'automates que nous considérons ici, les transitions entre les états de l'automate sont ainsi étiquetées par des événements. On distingue les événements observables $\mathcal{O} = \{o_1, ..., o_{no}\}$ et les événements non observables $\mathcal{U} = \{u_1, ..., u_{nu}\}$.

Une *trajectoire* du système décrit une évolution complète du système, elle correspond à une suite maximale (voire infinie) de transitions partant d'un état initial, elle est représentée sous la forme de la séquence d'événements correspondante [5]. Le système est modélisé ici comme un ensemble fini de trajectoires possibles.

Une trajectoire observable du système est obtenue par projection des trajectoires sur les événements observables et est une séquence σ (infinie ou non) d'événements observables. On fait l'hypothèse que le système n'est pas silencieux et donc que toute trajectoire contient au moins un événement observable.

Définition 1 (Observable)
On appelle observable *une séquence σ non vide d'événements observables de $\mathcal{O} = \{o_1, ..., o_{no}\}$ résultant de la projection d'une trajectoire du système. On note OBS l'ensemble fini des observables du système.*

Parmi les événements non observables, nous nous intéressons en particulier aux événements décrivant l'occurrence de fautes, dites fautes *élémentaires*, qui conduisent à un état fautif du système.

Un *mode de faute* caractérise l'état du système sous l'influence d'une ou plusieurs fautes élémentaires et est décrit par cet ensemble de fautes. Par extension, le mode correspondant à l'absence de faute, souvent appelé *mode nominal*, est inclus dans l'ensemble des modes du système et il est noté ok. Nous supposons dans cet article que le système ne peut pas subir de fautes multiples, et donc seule une faute au plus peut être présente dans le système à un instant donné. Ceci peut se justifier par le fait qu'une faute devrait être réparée avant qu'une nouvelle faute ne se produise. Dans ces conditions, on utilisera la notation f_i pour faire référence aussi bien à la faute élémentaire, à l'événement correspondant, ou au mode de faute dans lequel le système se trouve après l'occurrence de f_i. L'ensemble de tous les modes de fautes se résume alors à $\mathcal{F} = \{f_0, f_1, ..., f_{nf}\}$, avec $f_0 = ok$.

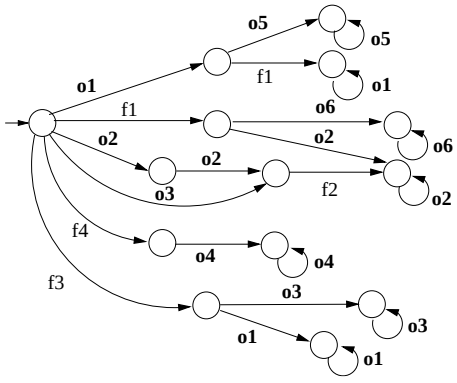

FIG. 1 – L'exemple illustratif.

Nous supposons aussi que les fautes sont permanentes, c'est-à-dire que le mode d'un système ne change pas sans réparation. Dire qu'une faute est présente dans un état signifie que cette faute a eu lieu avant d'arriver à cet état, et donc plus précisément que cet événement de faute appartient à toute trajectoire menant à cet état.

L'exemple suivant, que nous allons reprendre tout au long de cet article, illustre les concepts que nous venons de définir.

Exemple :

La figure 1 représente le modèle global d'un système à événements discrets. L'ensemble des fautes est $\mathcal{F} = \{ok, f_1, f_2, f_3, f_4\}$, les événements de faute f_i ne sont pas observables, et les autres événements o_1, o_2, o_3, o_4, o_5 et o_6 sont tous observables. [6]

Le système de l'exemple a 9 trajectoires et dispose de 7 observables. $o_1 o_5^\infty$ est une trajectoire mais est aussi l'observable de cette trajectoire ; elle est composée d'une observation o_1 suivie d'une séquence infinie de o_5. $o_2 o_2 f_2 o_2^\infty$ est une autre trajectoire produisant l'observable o_2^∞. $f_1 o_2^\infty$ est encore une autre trajectoire, dont on peut remarquer qu'elle correspond à exactement le même observable o_2^∞ : cela signifie que ces deux trajectoires ne pourront pas être discriminées à partir des observations effectuées.

Diagnostiquer qu'une faute a eu lieu de manière certaine revient à être capable de discriminer sans ambiguïté dans quel mode de faute le système se trouve. Mais cela n'est pas toujours possible. Nous appelons *macrofaute* un ensemble de fautes possibles. Une macrofaute décrit ainsi un état de croyance

[6] Dans cet exemple, nous faisons l'hypothèse, comme dans [15], que le système observé est "vivant", c'est-à-dire que toute trajectoire qui représente un comportement possible du système est infinie ainsi que l'observable σ associé à cette trajectoire. Cela n'a pas d'incidence sur la suite de cet article.

(ensemble d'hypothèses)[7].

Définition 2 (Macrofaute)
Une macrofaute F *est un ensemble de fautes élémentaires,* $F \subseteq \mathcal{F}, F \neq \emptyset$*. On dit que la macrofaute* F *est présente dans un état du système si et seulement si une des fautes élémentaires* $f_i \in F$ *est présente dans cet état (et seulement une, étant donnée l'absence de fautes multiples). On peut voir la macrofaute* F *comme la disjonction de fautes élémentaires la composant.*

Par exemple, la macrofaute $\{f_1, f_2\}$ représente soit la présence de f_1, soit la présence de f_2. La macrofaute $\{f_2, ok\}$ signifie la présence de f_2 ou l'absence de faute. Une macrofaute peut être un singleton ($F = \{f_i\}$), autrement dit, toute faute peut être vue comme une macrofaute particulière, généralisant le concept de faute. Un ensemble de macrofautes est noté E (avec $E \subseteq 2^{\mathcal{F}}$).

Si chaque élément de $\mathcal{F}$ appartient à au moins un élément de E alors l'ensemble E est un *ensemble couvrant* de $\mathcal{F}$.

Définition 3 (Ensemble couvrant)
Un ensemble de macrofautes E *est* couvrant *pour* $\mathcal{F}$ *si et seulement si* $\forall f_i \in \mathcal{F}, \exists F \in E$ *tel que* $f_i \in F$*. On dit que* E *couvre* $\mathcal{F}$*.*

Le concept de macrofaute est important puisque l'idée qui va être développée par la suite est qu'à défaut de pouvoir diagnostiquer parfaitement tout mode de faute possible, nous pouvons caractériser le *niveau de diagnosticabilité* du système en identifiant quelles macrofautes sont diagnosticables.

3.2 Réparations

Avant de donner une définition simplifiée de la notion de plan de réparation, suffisante pour notre propos, nous allons en illustrer de manière informelle les besoins par un petit exemple tiré du domaine des Web Services [18].

Sur la figure 2, à gauche, est représenté un workflow limité à une séquence de trois activités a_1, a_2, a_3 (un rectangle par activité). Une activité réalise une fonction dépendant des entrées de l'activité et fournissant des sorties. Les données d'entrée figurent en haut à gauche du rectangle, les données en sortie en bas à droite. On remarque que l'activité a_3 a besoin des données x_1 et x_2, produites respectivement par a_1 et a_2.

Les transitions qui conduisent de a_2 (respectivement a_3) à a_1 et a_2 symbolisent les possibilités offertes de réparation. Le plan r_1 (respectivement r_2) s'attache à compenser [8] les activités déjà effectuées pour se replacer dans la

[7] Un tel ensemble d'hypothèses correspond également à ce que doit considérer un diagnostic en ligne, qui va restreindre ces hypothèses au fur et à mesure des observations, jusqu'à, si possible, arriver à identifier une faute de manière certaine.

[8] La compensation annule les effets de l'activité et ramène le système à un état préalable à l'application de l'activité.

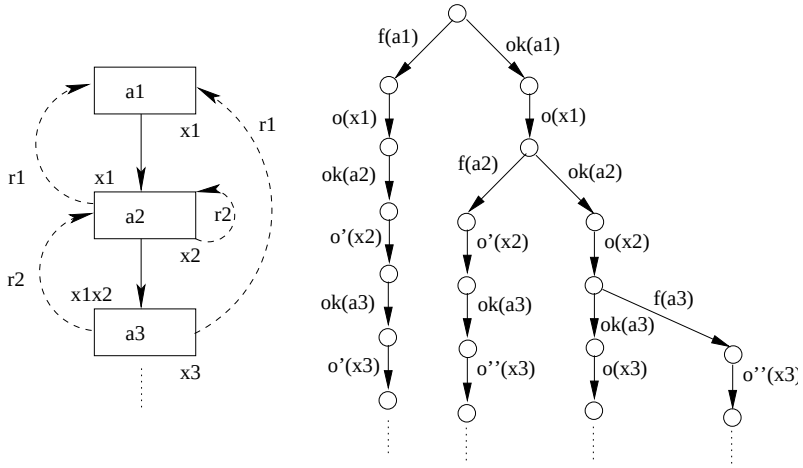

FIG. 2 – Réparer un worfklow.

situation où a_1 (respectivement a_2) peut être ré-exécutée, éventuellement en modifiant ses données d'entrée.

Sur la même figure, à droite, est représentée la partie des trajectoires de ce système afférent aux activités a_1, a_2, a_3 : une activité a_i en faute est représentée par un événement $f(a_i)$, une activité a_i non fautive est représentée par un événement $ok(a_i)$. Les événements du type $o(x_i), o'(x_i), o''(x_i)$ sont des événements de fin d'activités liés à la production de chaque activité [9] et de leur mode de fonctionnement. La production d'une activité dépend de son mode de fonctionnement et de ses entrées. Par exemple, l'activité a_3 en mode de fonctionnement normal peut produire une sortie $o'(x_3)$ différente de $o(x_3)$ car l'une de ses entrées x_1 et x_2 est incorrecte.

Supposons maintenant que seuls les événements du type $o(x_i), o'(x_i), o''(x_i)$ soient observables et que l'on observe $o(x_1)o'(x_2)o'(x_3)$, le système de diagnostic associe alors cette observation à la présence de la faute $f(a_1)$ et le plan r_1 peut être appliqué. De même si l'on observe $o(x_1)o'(x_2)o''(x_3)$, la faute $f(a_2)$ est présente et le plan r_2 peut être appliqué. Supposons maintenant que la sortie de l'activité a_3 n'est pas observable, les deux cas précédents se ramènent à l'observation de $o(x_1)o'(x_2)$ qui conduit à la présence de la macrofaute $\{f(a_1), f(a_2)\}$ et le plan r_1 peut être appliqué.

On voit ici que si l'on a au départ deux plans de réparation distincts pour les deux fautes, en fait le plan r_1 pour réparer $f(a_1)$ peut très bien s'appliquer pour réparer $f(a_2)$ (bien qu'il ne soit pas optimal) : lorsque l'on observe $o(x_1)o'(x_2)$, on sait que la macrofaute $\{f(a_1), f(a_2)\}$ a eu lieu, et on sait

[9] Dans le cas des Web Services, ces événements peuvent correspondre à la lecture des variables de sortie des activités ou encore à la levée d'exceptions.

que compenser jusqu'à a_1 et ré-exécuter permet de réparer ces deux fautes, il est donc possible d'exécuter ce plan r_1 même sans avoir discriminé les deux fautes possibles.

Ce simple exemple illustre toute la problématique de la prise en compte conjointe des capacités de diagnostic et de réparation. Il permet aussi de voir dans le cas d'une application spécifique à quoi peut ressembler un plan de réparation, comment il peut ramener le système dans un mode nominal, et comment un même plan de réparation peut s'appliquer à plusieurs fautes distinctes.

Pour autant, il est important de noter que le concept de plan de réparation est exogène au système supervisé. Selon l'application, il se traduira de diverses manières : si le système modifie une base de données lors de certaines transitions, une faute peut correspondre à une erreur d'enregistrement d'une donnée dans la base, et le plan consiste à aller simplement corriger cette donnée, le système poursuivant ensuite à partir de l'état courant ; comme dans notre exemple, la faute peut avoir invalidé certaines étapes, nécessitant de compenser d'abord ces étapes (en rétablissant par exemple les données initiales) et de replacer le système dans un état antérieur ; il peut même s'agir de décider de sauter plusieurs étapes futures devenues risquées, plaçant ainsi le système dans un état postérieur. Rien n'interdit que ce plan soit dynamique, dans le sens où les actions exogènes seront calculées en ligne en fonction de caractéristiques non prédictibles de la situation observée (par exemple la donnée effective corrompue). Mais dans ce travail, nous supposons bien sûr que le système supervisé (ses états et ses transitions) n'est pas modifié par la réparation, ce qui garantit de pouvoir étudier ses propriétés d'autoguérison lors de sa conception. De fait, ce plan exogène se ramène pour nous au niveau du système étudié à une simple modification de l'état courant du système, sans qu'il soit nécessaire pour cela d'ajouter de transitions spécifiques dans le système (dont le nombre peut être prohibitif, si par exemple une réparation consiste à se ramener à un état q_i, quel que soit l'état dans lequel on se trouve). Pour définir l'autoguérison, il suffit donc d'être capable de l'associer avec les fautes élémentaires à réparer : un plan de réparation r_k se caractérise comme une fonction, associant à tout état du système un nouvel état (appelé état-but) dans lequel le système se replace après exécution du plan. Un cas particulier est le plan vide, noté r_{ok}, qui consiste à laisser le système poursuivre en l'état, sans lancer aucune action exogène.

Il est important d'expliciter la relation existant entre plan de réparation et faute :

Définition 4 (Relation entre macrofautes et plans de réparation)
Soit l'ensemble des plans de réparation $\mathcal{R} = \{r_1, ..., r_{nr}\}$. $Répare(r_k, F)$ signifie que le plan de réparation r_k, s'il est appliqué dans un état dans lequel la macrofaute $F \in 2^{\mathcal{F}} \setminus \{\emptyset\}$ est présente, atteindra un état-but dans lequel aucune faute n'est présente.

Afin de généraliser au cas de l'état ok, nous considérons que le plan vide

est le seul plan applicable dans un état non fautif, autrement dit cela signifie que $Répare(r_{ok}, \{ok\})$.

Il faut remarquer que disposer d'un plan de réparation pour une macrofaute signifie que l'on dispose d'un plan capable de réparer le système alors que l'on ne sait pas quelle faute, parmi celles qui composent la macrofaute, a vraiment eu lieu. Ce plan peut être aussi utilisé pour réparer n'importe laquelle des fautes élémentaires incluses dans cette macrofaute (ce qu'exprime la propriété 1 ci-dessous). En général cependant, le plan de réparation associé à une macrofaute est plus coûteux (car il doit couvrir plus de cas) que le plan de réparation dédié spécifiquement à une faute élémentaire.

Propriété 1
$Répare(r_k, F)$ *ssi, pour toute faute* $f_i \in F$, $Répare(r_k, \{f_i\})$.

Un *plan de réparation* est supposé défini en fonction du contexte de l'application visée, de manière séparée du système observé. Nous avons supposé ci-dessus qu'il n'existait pas de condition restrictive d'applicabilité du plan, et qu'un plan peut s'appliquer dans toute situation où la macrofaute qu'il peut réparer est présente. L'extension à l'existence de conditions d'applicabilité du plan de réparation fait partie des travaux en cours. La définition 4 impose de n'appliquer le plan que dans les situations où la macrofaute est présente de manière certaine (une des fautes couvertes a effectivement eu lieu). Il est en effet en général très coûteux ou risqué d'appliquer un plan de réparation, par exemple le plan r_k tel que $Répare(r_k, F)$, dans un état où la faute (ici F) ne serait pas présente.

4 AUTOGUÉRISON

L'autoguérison peut se définir intuitivement comme suit :

Un système est autoguérissant si et seulement si, après l'occurrence d'une faute, un diagnostic peut être fait à la suite duquel un plan de réparation adapté à la situation est automatiquement déclenché.

Deux propriétés du système se cachent derrière cette définition, la diagnosticabilité et la réparabilité. Dans cette section, nous définissons ces deux propriétés et proposons ensuite une définition de l'autoguérison.

4.1 Diagnosticabilité

4.1.1 Nouvelle définition de diagnosticabilité

La diagnosticabilité repose sur la notion de *signature de fautes* [5]. De manière intuitive, une signature de faute est la mise en correspondance d'une faute avec l'ensemble de ses observables possibles.

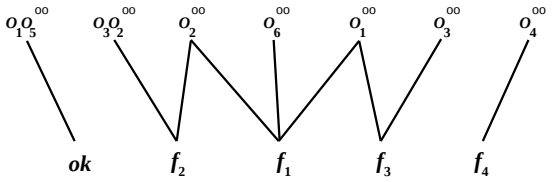

FIG. 3 – Relier les e-signatures aux fautes.

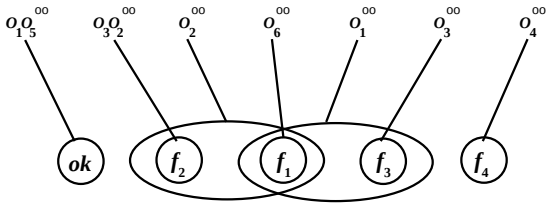

FIG. 4 – Macrofautes caractéristiques pour chaque observable.

Définition 5 (Relations entre fautes et observables)

- *La relation* $yields(f_i, \sigma)$ *exprime qu'il existe au moins une trajectoire contenant l'événement* $f_i \in \mathcal{F}$ *et correspondant à l'observable* $\sigma \in OBS$. σ *est appelé une signature élémentaire, ou* e-signature*, de la faute* f_i*, et* f_i *est appelée un* diagnostic *possible de l'observable* σ*. Par extension,* $\sigma \in OBS$ *est dite* e-signature *d'une macrofaute* F *si et seulement si* $\exists f_i \in F$ *telle que* $yields(f_i, \sigma)$*.*
- $MF(\sigma)$ *est la macrofaute (unique) contenant toutes les fautes qui peuvent produire* σ*, c'est-à-dire* $MF(\sigma) = \{f_i \mid yields(f_i, \sigma)\}$ *; en d'autres termes il s'agit de l'ensemble de tous les diagnostics possibles de* σ*. Il est important de remarquer que si* σ *est une e-signature d'une macrofaute* F*, alors cela signifie que* $F \cap MF(\sigma) \neq \emptyset$*.*
- MF *peut être généralisé aux ensembles de e-signatures :* $MF(\Sigma) = \bigcup_{\sigma \in \Sigma} MF(\sigma)$ *est l'ensemble de tous les diagnostics possibles des e-signatures présentes dans* Σ*.*

On a par exemple $MF(o_2^\infty) = \{f_1, f_2\}$ et $MF(o_1^\infty, o_2^\infty) = \{f_1, f_2, f_3\}$. Ces notations sont résumées dans les figures 3 et 4 : la première figure montre quels observables peuvent être reliés à quelles fautes, la deuxième identifie les ensembles MF en regroupant toutes les fautes constituant des diagnostics pour chaque observable.

La diagnosticabilité est classiquement définie pour une partition des fautes de $\mathcal{F}$ [5]. Dans ce travail nous ne considérons pas une partition de fautes mais un ensemble de macrofautes, lesquelles peuvent partager des fautes

élémentaires. Pour autant, afin de pouvoir discriminer les macrofautes, chacune doit pouvoir être associée à des observables distincts. De ce fait les ensembles d'observables correspondant aux macrofautes doivent toujours constituer une partition.

Ce qui amène à une nouvelle définition de la diagnosticabilité qui généralise la définition classique et qui est adaptée à l'étude de l'autoguérison.

Définition 6 (Diagnosticabilité d'un ensemble de macrofautes)
Un ensemble de macrofautes $E \subseteq 2^{\mathcal{F}}$ est diagnosticable*, ce qui est noté $Diagnosticable(E)$, si et seulement s'il existe une partition $\pi = \{\Sigma_1, \ldots, \Sigma_m\}$ des observables OBS telle que : $E = \{MF(\Sigma_j), j = 1 \ldots m\}$.*

Notons que de la définition 6 ci-dessus, il découle la propriété suivante :

Propriété 2
Tout ensemble de macrofautes E diagnosticable est couvrant.

Preuve : Si E est diagnosticable, les macrofautes de E regroupent les fautes élémentaires produisant les observables d'un même élément d'une partition des observables. Or chaque faute élémentaire appartient à au moins une trajectoire et produit donc au moins un observable (hypothèse faite dans la section 3.1), qui appartient par définition à la partition. E couvre donc bien (voir définition 3) l'ensemble des fautes élémentaires. □

On peut également observer que le cardinal de la partition π peut être strictement supérieur au cardinal de l'ensemble E lui correspondant : c'est le cas lorsqu'il existe au moins $\Sigma_i, \Sigma_j \in \pi, \Sigma_i \neq \Sigma_j$ tels que $MF(\Sigma_j) = MF(\Sigma_i)$.

Exemple : Poursuivons l'exemple précédent. Un premier ensemble trivial de macrofautes est $E_\top = \{\mathcal{F}\} = \{\{ok,\ f_1,\ f_2,\ f_3,\ f_4\}\}$ dans lequel les fautes ne sont pas discriminées : il est bien évidemment diagnosticable, et la partition correspondante est $\pi_\top = \{\{o_1o_5^\infty,\ o_1^\infty,\ o_2^\infty,\ o_3o_2^\infty,\ o_3^\infty,\ o_4^\infty,\ o_6^\infty\}\} = \{OBS\}$.
L'ensemble de macrofautes $E_1 = \{\{ok\}, \{f_1, f_2\}, \{f_1, f_3\}, \{f_4\}\}$ est diagnosticable avec $\pi_1 = \{\{o_1o_5^\infty\}, \{o_2^\infty, o_3o_2^\infty\}, \{o_1^\infty, o_3^\infty, o_6^\infty\}, \{o_4^\infty\}\}$. On peut remarquer que E_1 correspond aussi à une autre partition $\pi_2 = \{\{o_1o_5^\infty\}, \{o_2^\infty, o_3o_2^\infty, o_6^\infty\}, \{o_1^\infty, o_3^\infty\}, \{o_4^\infty\}\}$.
$E_2 = \{\{ok\}, \{f_1\}, \{f_2\}, \{f_3\}, \{f_4\}\}$ n'est pas diagnosticable parce qu'il y a des cas pour lesquels f_1 et f_2 ne peuvent être discriminées : il n'y a pas de partition d'observables qui puisse lui être associé. La même observation peut être faite pour f_1 et f_3.
La figure 5 illustre l'exemple de la partition π_2 et l'ensemble de macrofautes correspondant E_1.

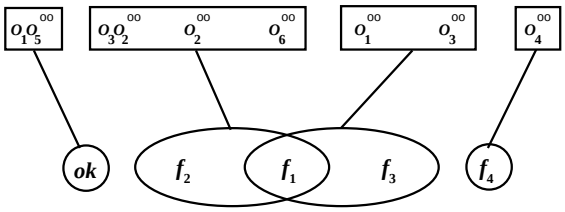

FIG. 5 – Exemple d'ensemble de macrofautes diagnosticable.

4.1.2 Niveau de diagnosticabilité

La définition 6 permet d'introduire la notion de *niveau de diagnosticabilité*. Supposons par exemple l'existence de deux ensembles diagnosticables distincts E_1 et E_2 tels que toute macrofaute de E_1 est, soit identique à une macrofaute de E_2, soit strictement incluse dans une macrofaute de E_2 et donc *plus petite* en terme de cardinalité. Si E_1 est diagnosticable et la macrofaute $F \in E_1$ n'est pas une macrofaute de E_2 alors il existe des situations observables dans lesquelles la présence de F peut être déterminée avec certitude, ce qui n'est pas établi dans E_2 qui détermine dans ces situations une macrofaute plus grande. À travers cet exemple, on voit donc que la diagnosticabilité de E_1 est plus *fine* que la diagnosticabilité associée à E_2. Par la suite, nous verrons que l'objectif est de déterminer si le système possède un niveau de diagnosticabilité compatible avec sa réparabilité. Le niveau de diagnosticabilité minimal correspond au cas où le seul ensemble de macrofautes diagnosticable est $E_\top = \{\mathcal{F}\}$, cela signifie que l'on ne peut discriminer aucun des modes de fautes. Un cas intéressant est celui, correspondant au niveau maximal théorique de diagnosticabilité, où l'ensemble des macrofautes réduites aux fautes élémentaires, $E_\bot = \{\{ok\}, \{f_1\}, \{f_2\}, \ldots \{f_{nf}\}\}$ est diagnosticable et on pourra toujours discriminer parfaitement chacune des fautes, ce qui correspond à la définition classique (voir [5]). Un autre cas intéressant correspondant à la détectabilité du système est celui où l'ensemble E est diagnosticable avec $\{ok\} \in E$ et aucune autre macrofaute de E ne contient $\{ok\}$. On est sûr dans ce cas de pouvoir détecter que le système n'est pas dans le mode nominal.

Par définition, un ensemble de macrofautes est diagnosticable si l'on peut lui associer une partition des observables. Il en découle que tous les ensembles diagnosticables peuvent être établis à partir de l'ensemble $\Pi(OBS)$ des partitions de OBS. L'ensemble $\Pi(OBS)$ constitue classiquement un treillis s'il est muni de la relation d'ordre partielle $\preceq$ definie par :

$$\forall \pi_1, \pi_2 \in \Pi(OBS), (\pi_1 \preceq \pi_2) \equiv (\forall e_1 \in \pi_1, \exists e_2 \in \pi_2, e_1 \subseteq e_2).$$

La partition π_1 est dite plus petite que la partition π_2 si tous ses élements

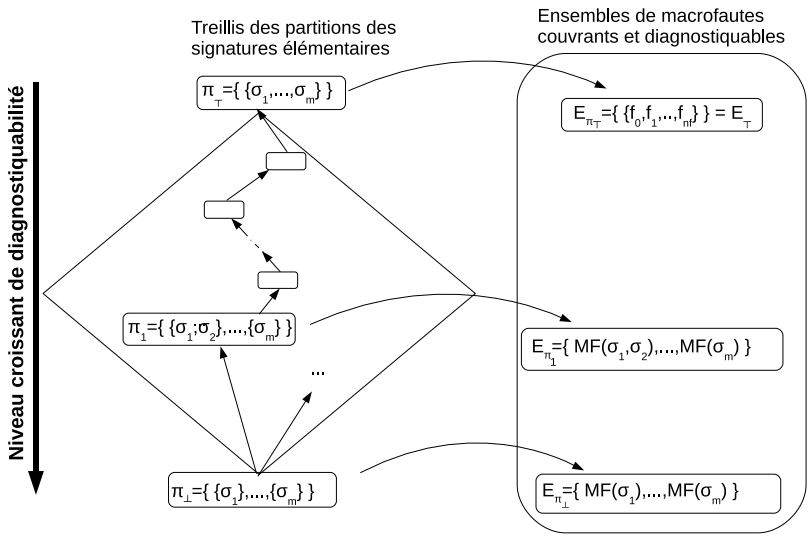

FIG. 6 – Treillis des partitions de signatures élémentaires et sa correspondance avec l'ensemble des ensembles couvrants et diagnosticables.

sont contenus dans les élements de π_2. Ainsi, le plus petit élément du treillis est $\pi_\perp = \{\{\sigma_1\}, \ldots, \{\sigma_m\}\}$, $\sigma_i \in OBS$ ($\forall \pi \in \Pi(OBS)$, $\pi_\perp \preceq \pi$) et le plus grand élément est l'ensemble $\pi_\top = \{\{\sigma_1, \ldots, \sigma_m\}\}$, ($\forall \pi \in \Pi(OBS)$, $\pi \preceq \pi_\top$) (voir Figure 6).

À partir de $\Pi(OBS)$, il est donc possible d'établir l'ensemble des ensembles diagnosticables $\mathcal{E}$. Si l'on note par E_π l'ensemble des macrofautes $E_\pi \triangleq \{MF(\Sigma_1), \ldots, MF(\Sigma_l)\}$ avec $\pi = \{\Sigma_1, \ldots, \Sigma_l\}$, alors on a :

$$\mathcal{E} = \{E_\pi, \pi \in \Pi(OBS)\}.$$

De plus, l'exploration de ce treillis en exploitant la relation d'ordre $\preceq$ permet d'explorer les ensembles diagnosticables avec des niveaux de diagnosticabilité décroissants.

Propriété 3
Soit π et π' deux partitions de $\Pi(OBS)$ telles que $\pi \preceq \pi'$ alors toute macrofaute de $E\pi$ est nécessairement incluse dans (ou identique à) une macrofaute de $E_{\pi'}$.

Preuve : Notons $\pi = \{\Sigma_1, \ldots, \Sigma_l\}$ et $\pi' = \{\Sigma'_1, \ldots, \Sigma'_h\}$. Comme $\pi \preceq \pi'$, il suit que $\forall i \in \{1, \ldots, l\}, \exists j \in \{1, \ldots, h\}, \Sigma_i \subseteq \Sigma'_h$, et donc finalement

$MF(\Sigma_i) \subseteq MF(\Sigma'_h)$, d'où le résultat. □

4.2 Réparabilité

La propriété de réparabilité découle directement de la relation entre macrofautes et plans de réparation. Ainsi, une macrofaute F est réparable si et seulement s'il existe un plan de réparation qui la répare.

Définition 7 (Réparabilité d'une macrofaute)
Une macrofaute F est réparable, noté $Réparable(F)$, ssi il existe un plan de réparation r_k tel que $Répare(r_k, F)$.

Nous pouvons déduire, grâce à la propriété 1, la propriété suivante :

Propriété 4
Si une macrofaute F est réparable, alors $\forall f_i \in F$, on a $Réparable(\{f_i\})$.

L'inverse n'est pas toujours vrai, puisque pour qu'une macrofaute soit réparable, il faut un plan commun à toutes les fautes la composant, alors que si toutes les fautes sont réparables, elles peuvent l'être par des plans différents.

La réparabilité d'un ensemble de macrofautes est alors définie comme la réparabilité de toutes les macrofautes de cet ensemble.

Définition 8 (Réparabilité d'un ensemble de macrofautes)
Un ensemble de macrofautes E (non vide) est réparable, ce qui est noté $Réparable(E)$, si et seulement si $\forall F \in E\ Réparable(F)$.

Nous allons voir que pour l'autoguérison, ce qui nous intéresse est d'associer diagnosticabilité et réparabilité via un ensemble de macrofautes E qui ait les deux propriétés, donc la définition ci-dessus nous suffit, et le fait qu'un système soit "globalement" réparable ou pas est secondaire. Nous pouvons néanmoins discuter ce point rapidement.

La nouvelle définition de diagnosticabilité revient à associer à un système un niveau de diagnosticabilité (qualifié de minimal lorsqu'aucune faute ne peut être discriminée). Selon cette définition, un système est désormais toujours, MAIS plus ou moins, diagnosticable. Il serait tout-à-fait possible de caractériser de la même manière le fait qu'une système soit *plus ou moins* réparable : plus grand est le nombre de fautes élémentaires pouvant être associées à un plan commun, plus le système est réparable (puisqu'il le sera même en l'absence de diagnostic discriminant ces fautes). Le niveau maximal est atteint lorsqu'il existe un plan "universel" réparant toutes les fautes élémentaires. Ces niveaux fonctionnent donc à l'inverse des niveaux de diagnosticabilité. A l'inverse, si les fautes élémentaires nécessitent toutes des plans distincts, le niveau de réparabilité sera minimal.

Qui plus est, au contraire de la diagnosticabilité, on peut voir qu'un système peut ne pas être du tout réparable, il suffit pour cela qu'une seule faute

élémentaire ne dispose d'aucun plan de réparation associé. Ces considérations conduisent aux définitions équivalentes suivantes pour la réparabilité d'un système.

Définition 9 (Réparabilité d'un système)
1. Un système est réparable ssi $\exists E$ *tel que* $Réparable(E)$ *et* E *est un ensemble couvrant.*
2. Un système est réparable ssi $\forall f_i \in \mathcal{F} Réparable(\{f_i\})$.

Propriété 5
Les deux définitions ci-dessus sont équivalentes.

Preuve : => Si il existe un ensemble de macrofautes réparable qui est couvrant, on en déduit, grâce à la propriété 4, que toute faute élémentaire est réparable.
<= Si toutes les fautes élémentaires sont réparables, alors il existe au moins un ensemble de macrofautes réparable et couvrant (celui où chaque macrofaute se réduit à une faute élémentaire). □
Exemple : Si le seul plan de réparation est r, avec $Répare(r, \{f_1, f_3\})$, on a $Réparable(\{f_1,f_3\})$, et aussi $Réparable(\{f_1\})$ et $Réparable(\{f_3\})$. Le système n'est pas réparable car les fautes f_2 et f_4 ne sont pas réparables.

4.3 Autoguérison

La définition d'autoguérison découle directement de celles de diagnosticabilité et de réparabilité.

Définition 10 (Ensemble autoguérissant de macrofautes)
Un ensemble de macrofautes E *est* autoguérissant *si et seulement si il est diagnosticable et réparable, i.e.* $Autoguérissant(E)$ *ssi* $Diagnosticable(E)$ *et* $Réparable(E)$

Étant donné un ensemble autoguérissant E, l'observation de $\sigma \in OBS$ est toujours associée à une macrofaute $F_\sigma \in E$ puisque E est diagnosticable. De plus, puisque E est réparable, il existe un plan de réparation pour F_σ (plan de réparation qui répare toute faute élémentaire de F_σ), donc F_σ peut être réparée.

Définition 11 (Système autoguérissant)
Un système est autoguérissant *si et seulement si il existe un ensemble autoguérissant* E*, i. e. ssi* $\exists E$ *tel que* $Autoguérissant(E)$.

On remarquera ici aussi qu'il n'est pas nécessaire d'imposer que E soit un ensemble couvrant, il l'est nécessairement à partir du moment où il est diagnosticable (propriété 2).
Exemple :

– Cas 1 : $R\acute{e}parable(\{ok\})$ et $R\acute{e}parable(\{f_1, f_3\})$ et $R\acute{e}parable(\{f_1, f_2\})$ et $R\acute{e}parable(\{f_4\})$ L'ensemble $E_1 = \{\{ok\}, \{f_1, f_2\}, \{f_1, f_3\}, \{f_4\}\}$, dont nous avons déjà montré qu'il était diagnosticable, est de ce fait réparable. Le système est donc autoguérissant.
– Cas 2 : $R\acute{e}parable(\{ok\})$ et $R\acute{e}parable(\{f_1, f_3\})$ et $R\acute{e}parable(\{f_2\})$ et $R\acute{e}parable(\{f_4\})$
Aucun ensemble de macrofautes n'est autoguérissant puisqu'il n'existe pas de plan de réparation pour $\{f_1, f_2\}$. Le système n'est donc pas autoguérissant.

4.4 Propriétés de l'autoguérison

Nous énonçons un théorème important qui permettra par la suite dans la section 5 de proposer un algorithme de vérification de la propriété d'autoguérison.

Soit $E_{\pi_\perp}$ l'ensemble diagnosticable de macrofautes associé à la partition la plus fine du treillis $\Pi(OBS)$, on a alors le résultat suivant.

Théorème 1
Le système est autoguérissant si et seulement si l'ensemble $E_{\pi_\perp}$ est réparable.

Ce théorème met en relief un ensemble diagnosticable particulier dont il suffit de déterminer la réparabilité pour déterminer si le système est autoguérissant. Ce théorème dit en effet que si cet ensemble $E_{\pi_\perp}$ n'est pas réparable alors aucun ensemble diagnosticable possible ne l'est. La démonstration de ce résultat nécessite tout d'abord la démonstration du lemme suivant.

Lemme 1
Soient $\pi, \pi' \in \Pi(OBS)$ tels que $\pi \preceq \pi'$. Si E_π n'est pas réparable alors $E_{\pi'}$ n'est pas réparable.

Preuve (lemme) : Si E_π n'est pas réparable, alors il existe une macrofaute F de E_π qui n'est pas réparable. Comme $\pi \preceq \pi'$, la propriété 3 implique qu'il existe dans $E_{\pi'}$ une macrofaute F' telle que $F \subseteq F'$. Par conséquent, F' n'est pas réparable (Définition 7) d'où le résultat. □

Ce lemme exprime formellement la relation entre le niveau de diagnosticabilité et la réparabilité. Pour un certain niveau de diagnosticabilité, si l'ensemble E_π n'est pas réparable alors tout ensemble $E_{\pi'}, \pi \preceq \pi'$ dont le niveau de diagnosticabilité est plus faible n'est pas réparable non plus. Démontrons maintenant le théorème 1.

Preuve (théorème 1) : Démontrons d'abord que si l'ensemble $E_{\pi_\perp}$ est réparable alors le système est autoguérissant. Par construction, $E_{\pi_\perp}$ est un ensemble diagnosticable, il est donc possible d'exhiber dans le système un

ensemble diagnosticable et réparable : le système est autoguérissant par définition.

Démontrons maintenant que si le système est autoguérissant alors $E_{\pi_\perp}$ est réparable. $\pi_\perp$ est la partition la plus fine du treillis $\Pi(OBS)$: $\forall \pi \in \Pi(OBS), \pi_\perp \preceq \pi$. Tout ensemble diagnosticable E étant associé à au moins une partition π de $\Pi(OBS)$ (c.-à-d. il existe π tel que $E = E_\pi$), si $E_{\pi_\perp}$ n'est pas réparable, alors aucun ensemble diagnosticable n'est réparable (Lemme 1), d'où le résultat par contraposée. □

5 VÉRIFIER LA CAPACITÉ D'AUTOGUÉRISON

Vérifier la capacité d'autoguérison revient à déterminer un ensemble couvrant de macrofautes qui soit à la fois diagnosticable et réparable. Comme nous l'avons vu, tout ensemble diagnosticable peut être associé au moins à une partition π du treillis $\Pi(OBS)$, l'ensemble de macrofautes étant alors noté E_π. Vérifier l'autoguérison revient alors à trouver une partition π de $\Pi(OBS)$ telle que $E(\pi)$ soit réparable.

Un moyen simpliste de procéder est donc de tester chaque ensemble diagnosticable pour déterminer s'il est réparable tant que l'on en a pas trouvé un. Mais cette solution, qui dérive directement de la définition de l'autoguérison n'est pas satisfaisante car elle est trop coûteuse. Il existe une solution beaucoup plus efficace qui découle directement du théorème 1 de la section précédente et ne nécessite qu'un seul test de réparabilité.

5.1 Algorithme

L'algorithme que nous proposons maintenant va donc se contenter de vérifier que $E_{\pi_\perp}$ est réparable. Ce qui revient à chercher des plans de réparations applicables pour chaque macrofaute $MF(\sigma) \in E_{\pi_\perp}$.

Notation 1
$RP(f_i)$ est l'ensemble des plans de réparation r_k tel que $Répare(r_k, f_i)$.
$AP(\sigma)$ est l'ensemble des plans capables de réparer les fautes ayant pour signature élémentaire σ ; il peut être défini comme l'intersection des ensembles de plans qui réparent toute faute élémentaire dont σ est une signature élémentaire : $AP(\sigma) = \bigcap_{f_i \in MF(\sigma)} RP(f_i)$.

L'algorithme vérifiant la capacité d'autoguérison est le suivant :

for all $\sigma \in OBS$ **do**
  Calculer $AP(\sigma) = \bigcap_{f_i \in MF(\sigma)} RP(f_i)$
  if $AP(\sigma) = \emptyset$ **then**
    retourne(Pas Autoguérissant) ; fin
  end if
end for

retourne(Autoguérissant) ; fin

En d'autres termes, l'algorithme conclut que le système n'est pas autoguérissant si et seulement s'il existe une signature élémentaire σ pour laquelle $AP(\sigma)$ est vide.

	$o_1o_5^\infty$	o_1^∞	o_2^∞	$o_3o_2^\infty$	o_3^∞	o_4^∞	o_6^∞
ok	r_{ok}						
f_1		r_1, r_2	r_1, r_2				r_1, r_2
f_2			r_1, r_3	r_1, r_3			
f_3		r_2, r_3			r_2, r_3		
f_4						r_4	
	$\mathbf{r_{ok}}$	$\mathbf{r_2}$	$\mathbf{r_1}$	$\mathbf{r_1/r_3}$	$\mathbf{r_2/r_3}$	$\mathbf{r_4}$	$\mathbf{r_1/r_2}$

FIG. 7 – Algorithme appliqué à l'exemple.

Exemple : L'algorithme est illustré sur l'exemple de la figure 1 par le tableau de la figure 7. Chaque colonne représente une des sept signatures élémentaires appartenant à la partition $\pi_\perp$. L'ensemble de macrofautes diagnosticables correspondant est $E_{\pi_\perp} = \{\{ok\}, \{f_1, f_3\}, \{f_1, f_2\}, \{f_2\}, \{f_3\}, \{f_4\}, \{f_1\}\}$ Chaque ligne est associée à une faute élémentaire et l'on suppose dans cet exemple qu'il existe quatre plans de réparations :

$RP(ok) = \{r_{ok}\}$
$RP(f_1) = \{r_1, r_2\}$
$RP(f_2) = \{r_1, r_3\}$
$RP(f_3) = \{r_2, r_3\}$
$RP(f_4) = \{r_4\}$

Pour chaque cellule de ligne f_i et de colonne σ, l'algorithme la marque comme étant active si σ est une signature élémentaire de f_i, puis il y ajoute l'ensemble des plans réparant la faute f_i. Une fois le tableau rempli, il suffit d'attribuer à chaque colonne les plans qui se retrouvent dans toutes les cellules actives de la colonne (ce résultat est donné par la dernière ligne du tableau). Dans cet exemple, aucun des ensembles résultants n'est vide ce qui signifie que $E_{\pi_\perp}$ est réparable et donc que le système est autoguérissant.

5.2 Extensions

Cet algorithme de vérification peut être étendu afin de fournir une stratégie complète pour la réparation. Comme on peut le constater dans le tableau illustrant l'algorithme sur l'exemple, chaque colonne est associée à un ensemble de plans possibles. Ces plans peuvent être classés selon la qualité de l'état de bon fonctionnement retrouvé. Le meilleur plan sera alors sélectionné pour chaque colonne : cette association directe entre signatures élémentaires et plans à appliquer peut alors "oublier" les fautes sous-jacentes. De plus, en fusionnant les colonnes correspondant au même plan choisi, on construit de

facto une nouvelle partition π et un nouvel ensemble de macrofautes correspondant $E(\pi)$ diagnosticable et réparable.

Dans l'exemple de la figure 7, on peut choisir les plans r_1 pour les observables $o_3o_2^\infty$ et o_6^∞, et r_2 pour l'observable o_3^∞, ce qui conduit à $\pi = \{\{o_1o_5^\infty\}, \{o_2^\infty, o_3o_2^\infty, o_6^\infty\}, \{o_1^\infty, o_3^\infty\}, \{o_4^\infty\}\}$ et $E_\pi = \{\{ok\}, \{f_1, f_2\}, \{f_1, f_3\}, \{f_4\}\}$, qui correspond à l'ensemble de plans de réparation $\{r_{ok}, r_1, r_2, r_4\}$. Alors que si l'on choisit r_3 pour les deux signatures $o_3o_2^\infty$ et o_3^∞, et r_2 pour o_6^∞, il en résulte $\pi = \{\{o_1o_5^\infty\}, \{o_2^\infty\}, \{o_1^\infty, o_6^\infty\}, \{o_3o_2^\infty, o_3^\infty\}, \{o_4^\infty\}\}$ et $E_\pi = \{\{ok\}, \{f_1, f_2\}, \{f_1, f_3\}, \{f_2, f_3\}, \{f_4\}\}$ correspondant à l'ensemble de plans $\{r_{ok}, r_1, r_2, r_3, r_4\}$.

En d'autres termes, lorsque le système est autoguérissant, on va pouvoir se déplacer dans le treillis de $\pi_\perp$ vers π, avec $\pi_\perp \preceq \pi$, en acceptant un niveau de diagnosticabilité plus faible et des plans de réparation plus génériques, éventuellement plus coûteux, tant que l'ensemble correspondant E_π reste réparable.

6 FORMES AFFAIBLIES D'AUTOGUÉRISON

L'autoguérison telle que définie précédemment est parfaitement appropriée pour vérifier un système puisqu'elle indique que chaque faute élémentaire peut toujours être réparée, soit parce qu'elle est elle-même diagnosticable et réparable, soit parce que c'est le cas d'une macrofaute la couvrant. En d'autres termes, lors de l'occurrence d'une faute, les observations et plans de réparation sont suffisants pour garantir un diagnostic permettant de déclencher un plan de réparation adapté.

Cependant, identifier un sous-ensemble autoguérissant lorsque le système entier ne l'est pas peut aussi revêtir un intérêt : en phase de conception d'un système, on peut alors guider le concepteur pour qu'il améliore la capacité d'autoguérison existante.

Dans la suite, nous définissons deux formes affaiblies d'autoguérison. La première, appelée *autoguérison faible*, consiste à garantir que chaque faute peut être diagnostiquée et réparée au moins dans certains cas (pour certaines trajectoires). La seconde, appelée *autoguérison partielle*, consiste à garantir qu'il existe un sous-ensemble de fautes qui sont dans tous les cas autoguérissantes.

6.1 Autoguérison faible

6.1.1 Diagnosticabilité faible

L'autoguérison faible s'appuie sur une simple modification de la définition de diagnosticabilité, où l'on autorise certains observables à ne pas être associés à des macrofautes de l'ensemble E (dans la définition ci-dessous, ils sont regroupés dans un élément supplémentaire de la partition des observables nommé Σ_{out}) : ce sont les observables qui peuvent être produits par

plusieurs des macrofautes de E, et qui mettent donc en échec la capacité de discriminer ces macrofautes. Pour autant, pour chaque macrofaute de E, il doit subsister au moins un observable qui lui est exclusivement associé (ceci est garanti par la définition d'une partition car Σ_j est nécessairement non vide). De plus, on doit exiger que E soit couvrant afin d'assurer qu'il existe au moins une trajectoire pour chaque faute élémentaire permettant l'autoguérison.

Définition 12 (Diagnosticabilité faible d'un ensemble de macrofautes)
L'ensemble E est faiblement diagnosticable*, ce qui est noté $FDiagnosticable(E)$, si et seulement s'il existe une partition $\pi = \{\Sigma_1, \ldots, \Sigma_m, \Sigma_{out}\}$ des observables OBS telle que : $E = \{MF(\Sigma_j), j = 1 \ldots m\}$ et E est couvrant.*

On peut remarquer que si l'ensemble de macrofautes E est diagnosticable ($Diagnosticable(E)$) alors il est aussi faiblement diagnosticable et donc on a $FDiagnosticable(E)$.

Exemple :
$E_\top$ et E_1 sont faiblement diagnosticables puisqu'ils sont diagnosticables. $E_2 = \{\{ok\}, \{f_1\}, \{f_2\}, \{f_3\}, \{f_4\}\}$ est faiblement diagnosticable si on lui associe la partition $\pi_{2out} = \{\{o_1 o_5^\infty\}, \{o_6^\infty\}, \{o_3 o_2^\infty\}, \{o_3^\infty\}, \{o_4^\infty\}, \{o_1^\infty, o_2^\infty\}\}$, où chaque macrofaute a au moins un observable associé de manière discriminante, et où $\Sigma_{out} = \{o_1^\infty, o_2^\infty\}$ contient tous les observables "ambigus" vis-à-vis des macrofautes de E_2.

6.1.2 Autoguérison faible

La définition d'autoguérison faible suit :

Définition 13 (Autoguérison faible)
L'ensemble de macrofautes E est faiblement autoguérissant*, ce qui est noté $FAutoguérissant(E)$, si et seulement si il est faiblement diagnosticable et réparable.*
Le système est faiblement autoguérissant si et seulement si il existe E tel que $FAutoguérissant(E)$.

Exemple :
Sachant que $Réparable(\{ok\})$ et $Réparable(\{f_1\})$ et $Réparable(\{f_2\})$ et $Réparable(\{f_3\})$ et $Réparable(\{f_4\})$, E_2 n'est pas autoguérissant car il n'est pas diagnosticable, mais il est faiblement autoguérissant car il est faiblement diagnosticable et réparable. Par conséquent, le système n'est pas autoguérissant mais faiblement autoguérissant.

6.2 Autoguérison partielle

Une autre forme d'autoguérison, dite partielle, s'avère intéressante lorsque certaines fautes ne sont pas réparables. On aimerait alors pouvoir vérifier qu'il existe un sous-ensemble de macrofautes diagnosticables E_p couvrant des fautes élémentaires réparables dans toutes les situations où elles sont possiblement présentes, i.e. pour toutes les situations où la faute peut se trouver dans les hypothèses de diagnostic. Pour cela, il faut que les fautes couvertes par E_p ne sont pas présentes dans des macrofautes non réparables.

Pour définir cela de manière générique, on commence par définir le fait que deux ensembles de macrofautes soient disjoints au sens des fautes élémentaires couvertes, ce que nous appellerons ensembles *séparables* :

Définition 14 (Sous-ensembles de macrofautes séparables)
Soient deux sous-ensembles de macrofautes E et E'. On dira qu'ils sont séparables *si et seulement si $\forall F \in E, \forall F' \in E', F \cap F' = \emptyset$.*

L'autoguérison partielle suit immédiatement :

Définition 15 (Autoguérison partielle)
L'ensemble de macrofautes E_p est partiellement autoguérissant*, ce qui est noté $PAutoguérissant(E_p)$, si et seulement si il existe E tel que*
- $E_p \subseteq E$
- *E_p et E_{NR} sont séparables, avec $E_{NR} = \{F, F \in E$ et $non(Réparable(F))\}$*
- $Diagnosticable(E)$
- $Réparable(E_p)$

On peut remarquer que tout ensemble de macrofautes autoguérissant est aussi partiellement autoguérissant.

Exemple :
L'ensemble réparable $E_{1p} = \{\{Ok\}, \{f_1, f_2\}, \{f_1, f_3\}\}$ est partiellement autoguérissant dans le cas où l'on a $Réparable(\{f_1, f_2\})$ et $Répa\,rable(\{f_1, f_3\})$ mais $non(Réparable(\{f_4\}))$. En effet, $E_{1p} \subset E_1$, E_{1p} et $E_{NR} = \{\{f_4\}\}$ sont séparables, et E_1 est diagnosticable. On peut noter que $E_{2p} = \{\{ok\}, \{f_1, f_2\}\}$ dans ce cas est aussi partiellement autoguérissant. Bien entendu, en pratique, on s'intéressera plus particulièrement aux ensembles partiellement autoguérissant "maximaux" tels E_{1p}.

Comme nous l'avons suggéré, l'autoguérison partielle est la seule qui soit compatible avec une propriété de réparation partielle du système, c'est-à-dire l'existence d'un ensemble de macrofautes non nécessairement couvrant qui soit réparable :

Définition 16 (Réparabilité partielle du système)
Un système est partiellement réparable ssi il existe un ensemble de macrofautes E tel que $Réparable(E)$

Pour résumer, à partir des définitions précédentes, on obtient :

Propriété 6

Tout système réparable est partiellement réparable
Tout système autoguérissant est réparable
Tout système faiblement autoguérissant est réparable
Tout système partiellement autoguérissant est partiellement réparable.

Un système qui est autoguérissant ou faiblement autoguérissant doit nécessairement être (complètement) réparable.

Pour résumer, un ensemble de macrofautes autoguérissant est tel que pour toute faute on saura toujours associer un plan de réparation, via une macrofaute contenant la faute. Un ensemble faiblement autoguérissant est tel que pour toute faute, il existe des situations dans lesquelles on saura réparer, mais il peut aussi exister des situations pour lesquelles on ne saura pas trouver de plan applicable. Enfin, un ensemble de macrofautes partiellement autoguérissant est tel que l'on saura associer un plan de réparation dans toutes les situations pour un sous-ensemble des fautes, mais pas pour toutes.

D'autres formes affaiblies d'autoguérison pourraient être définies, notamment en couplant l'autoguérison faible et partielle.

7 FACILITER LA CONCEPTION D'UN SYSTÈME AUTOGUÉRISSANT

En contrepoint de l'étude théorique qui vient d'être menée sur les formes affaiblies de la capacité d'autoguérison, un concepteur aura généralement à faire face à la préoccupation plus prosaïque de ce qu'il convient de faire en présence d'un système non autoguérissant... Il serait alors judicieux de l'aider à trouver des moyens d'améliorer les caractéristiques de son système afin de le rendre autoguérissant. Nous allons montrer rapidement que les structures que nous venons de mettre en place pour la vérification de l'autoguérison peuvent être avantageusement utilisées pour prendre les bonnes décisions.

La question est donc : que faire lorsque, après avoir exécuté l'algorithme, l'un des ensemble de plans applicables $AP(\sigma)$ est vide ? En premier lieu, il convient de modifier l'algorithme pour le forcer à itérer sur toutes les signatures élémentaires au lieu de s'arrêter à la première colonne pour laquelle $AP(\sigma) = \emptyset$ ... Ensuite, chaque colonne pour laquelle $AP(\sigma) = \emptyset$ indique qu'il n'y a aucun plan réparant toutes les fautes élémentaires présentes dans $MF(\sigma)$. Une décision possible, si tant est qu'elle soit possible, est justement de concevoir un nouveau plan qui serait à même de réparer indistinctement l'ensemble de ces fautes. Une autre possibilité est d'augmenter l'observabilité du système, par exemple grâce à de nouveaux moniteurs, pour récupérer

des observations supplémentaires qui permettront de distinguer les trajectoires correspondant aux fautes élémentaires incriminées.

Dans notre exemple, supposons que r_2 ne répare pas f_1. Alors aucun plan ne peut être associé à o_1^∞. Augmenter les capacités de réparation revient ici à construire un plan r_5 capable de réparer aussi bien f_1 que f_3. Augmenter à l'opposé les capacités d'observation exige de produire et/ou de détecter de nouveaux événements sur au moins l'une des deux trajectoires $o_1 f_1 o_1^\infty$ et $f_3 o_1 o_1^\infty$ de la figure 1, afin que leurs projections sur les observables diffèrent. Par exemple la seconde trajectoire pourrait devenir $f_3 o_1 o_6 o_1^\infty$, faisant apparaître un nouvel événement observable et donc une nouvelle e-signature $o_1 o_6 o_1^\infty$, qui pourrait être associée à r_2 ou r_3, o_1^∞ étant dès lors associée au plan r_1 seul.

Le résultat intéressant que nous souhaitons mettre en avant est que notre algorithme fournit également un outil facilitant l'identification des actions qui peuvent être entreprises, au niveau de la conception du système, pour assurer sa capacité d'autoguérison.

8 CONCLUSION

La principale contribution de cet article est de proposer une définition formelle et intégrée de la capacité d'*autoguérison* d'un système dynamique, qui s'appuie sur les capacités de diagnostic et de réparation. Il est intéressant de constater qu'il n'est pas nécessaire que le système soit diagnosticable vis-à-vis de chacune des fautes élémentaires, ni de disposer de plans de réparation distincts, spécialisés, pour chaque faute. Il convient plutôt de rechercher la présence d'un ensemble "suffisant" de situations diagnosticables pouvant être associés à un ensemble "suffisant" de réparations disponibles. Pour autant que nous sachions, une telle définition n'avait jusqu'alors pas été proposée.

Bien entendu ce travail est une première étape qui se doit d'être prolongée de plusieurs manières. À l'heure actuelle, nous souhaitons appliquer ces résultats au domaine d'application suggéré dans l'introduction, à savoir les services web, dans le cadre du projet européen WS-DIAMOND [18], dans lequel nous étudions un certain nombre d'extensions permettant d'aborder des exemples plus sophistiqués et réalistes, concernant pour l'essentiel les propriétés et les conditions d'applicabilité des plans de réparation :

– Conditions temporelles : un premier et important défi est de prendre en compte le délai nécessaire à l'obtention d'un diagnostic, provenant de la nécessité d'attendre d'avoir un nombre suffisant d'observations ; ce délai peut entrer en conflit avec un délai de péremption associé aux plans de réparation, au-delà duquel il sera trop tard pour pouvoir appliquer le plan (par exemple parce que certains effets de la faute ne peuvent plus être compensés). Plus généralement, il convient de caractériser les observations qui rendent valide ou caduque un plan, et de les comparer

temporellement aux observations qui valident un diagnostic, ces informations étant alors intégrées dans notre algorithme, afin de ne relier un plan à une faute que s'il est encore applicable lorsque le diagnostic est disponible.
– Fautes multiples : si f_2 est susceptible de se produire quelque temps après f_1, il peut s'avérer plus judicieux d'attendre et d'appliquer un plan qui réparera les deux fautes que d'appliquer deux plans successifs... ?
– Stratégies d'autoguérison : les deux points précédents suggèrent également la nécessité plus générale d'optimisation dynamique de la réparation, en partant du principe qu'un système potentiellement fautif évolue entre des états de diagnostic partiel, dans lesquels on peut soit appliquer un plan de réparation, soit attendre un raffinement du diagnostic. Il s'agit alors de comparer ces stratégies afin de pouvoir prendre les "meilleures" décisions en ligne, ce qui requiert un raisonnement en termes d'utilité des décisions de réparation.

RÉFÉRENCES

[1] M. Basseville. On fault detectability and isolability. *European Journal of Control*, 7(8) :625–637, 2001.

[2] J. Chen et R.J. Patton. A re-examination of fault detectability and isolability in linear dynamic systems. In *Proceedings of the 2nd IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (Safeprocess'94)*, pages 590–596, Helsinki, Finland, 1994.

[3] A. Cimatti, C. Pecheur et R. Cavada. Formal verification of diagnosability via symbolic model checking. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 363–369, Acapulco, Mexico, 2003.

[4] O. Contant, S. Lafortune et D. Teneketsis. Diagnosis of intermittent faults. *Journal of Discrete Event Dynamic Systems : Theory and Applications*, 14 :171–202, 2004.

[5] M.-O. Cordier, L. Travé-Massuyès et X. Pucel. Comparing diagnosability in continous and discrete-event systems. In *Proceedings of the 17th International Workshop on Principles of Diagnosis, (DX'06)*, pages 55–60, 2006.

[6] G. Garcia, J. Bernussou et D. Arzelier. Robust stabilization of discrete-time linear systems with norm-bounded time-varying uncertainty. *Systems & Control Letters archive*, 22,5 :327–339, 1994.

[7] D. Ghosh, R. Sharman, H. R. Rao et S. Upadhyaya. Self-healing systems : survey and synthesis. *Decision Support Systems*, 42(4) :2164–2185, 2007.

[8] T. Jéron, H. Marchand, S. Pinchinat et M-O. Cordier. Supervision patterns in discrete event systems diagnosis. In *Proceedings of the Workshop on Discrete Event Systems (WODES'06)*, Ann-Arbor, USA, 2006.

[9] S. Jiang, Z. Huang, V. Chandra et R. Kumar. A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46(8) :1318–1321, 2001.

[10] Y. Pencolé. Assistance for the design of a diagnosable component-based system. In *Proceedings of the 17th International Conference on Tools with Artificial Intelligence (ICTAI 05)*, pages 549–556. IEEE Computer Society, 2005.

[11] X. Pucel, S. Bocconi, C. Picardi, D. Theseider Dupre et L. Travé-Massuyès. Diagnosability analysis for web services with constraint-based models. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX'07)*, pages 360–367, Nashville, USA, 2007.

[12] P.J.G. Ramadge et W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 17,1 :81–98, 1989.

[13] A. Saboori et S. H. Zad. Fault recovery in discrete event systems. In *Proceedings of the ICSC congress on Computational Intelligence Methods and Applications (CIMA'05)*, Istanbul, Turkey, 2005.

[14] G. Salaün, L. Bordeaux et M. Schaerf. Describing and reasoning on web services using process algebra. In *Proceedings of the International conference on Web Services (ICWS'04)*, pages 43–51, San Diego, USA, 2004. IEEE Computer Society Press.

[15] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen et D. Teneketzis. Diagnosability of discrete event system. *IEEE Transactions on Automatic Control*, 40(9) :1555–1575, 1995.

[16] A. Schumann et Y. Pencolé. Scalable diagnosability checking of event-driven system. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 575–580, Hyderabad, India, 2007.

[17] M. Shaw. Self-healing : Softening precision to avoid brittleness. In *Proceedings of the First Workshop on Self-Healing Systems (WOSS'02)*, pages 111–113, 2002.

[18] The Ws-DIAMOND team. Ws-DIAMOND : Web services DIagnosability, MOnitoring and DIagnosis. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX'07)*, pages 243–250, Nashville, USA, 2007.

[19] L. Travé-Massuyès, T. Escobet et R. Milne. Model-based diagnosability and sensor placement application to a frame 6 gas turbine subsystem. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, volume 1, pages 551–556, 2001.

[20] L. Travé-Massuyès, T. Escobet et X. Olive. Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Transactions on Systems, Man and Cybernetics, Part A : Systems and Humans*, 36(6) :1146–1160, 2006.

[21] R. Washington, K. Golden et J. Bresina. Plan execution, monitoring, and adaptation for planetary rovers. In *Proceedings of the IJCAI'99 Workshop on Scheduling and Planning meet Real-time Monitoring in a Dynamic and Uncertain World*, Stockholm, Sweden, 1999.

[22] Y. Yan, Y. Pencolé, M.-O. Cordier et A. Grastien. Monitoring web service networks in a model-based approach. In *Proceedings of the 3rd European Conference on Web Services (ECOWS'05)*, Växjö, Sweden, 2005.

[23] T. Yoo et S. Lafortune. Polynomial-time verification of diagnosability of partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 47(9) :1491–1495, 2002.