# Diagnostic distribué à base de modèles sans calcul préalable des conflits

**Vincent Armant Philippe Dague Laurent Simon**

Université Paris Sud (LRI)& CNRS - INRIA (Saclay)
4 rue Jacques Monod, Parc Club 91400 Orsay
{armant,dague,simon}@lri.fr

**Résumé**

Il existe plusieurs moyens pour parer aux défaillances d'un système distribué, dans lequel tous les pairs travaillent pour un même but, en suivant le même protocole. On peut ainsi mettre en place un mécanisme de duplication des processus et des données, mais, avec l'émergence des services webs, le besoin de confidentialité et l'augmentation croissante du volume des données, une telle solution n'est pas toujours envisageable. Certains problèmes demandent par exemple de permettre à tous les pairs du système de détecter dès que possible la défaillance éventuelle d'un autre pair. Dans cet article, nous étendons le principe bien connu du diagnostic à base de modèles aux systèmes d'inférence distribués, dans lequel le réseau est bâti de "proche en proche". Par rapport aux autres approches de diagnostic distribué à base de modèles, nous calculons tous les diagnostics minimaux de façon anytime, sans passer par les conflits.

**Abstract**

A lot of methods exists to prevent errors and incorrect answers in a distributed framework, where all peers work together for the same purpose, under the same protocol. For instance, one may limit them by replication of data and processes among the network. However, with the emergence of web services, the willing for privacy, and the constant growth of data size, such a solution may not be applicable. On some problems, failure of a peer has to be detected by the whole system. In this paper, we propose an approach to diagnose abnormal behaviors of the whole system by extending the well known model-based diagnosis framework to a fully distributed inference system, where each peer only knows the existence of its neighbors. Contrasting to previous works on model-based diagnosis, our approach computes all minimal diagnoses in an anytime way, without needs to get any conflict first.

## 1 Introduction

Le diagnostic à base de modèles a été introduit dans les années 80 par [13, 8] et a, depuis, été largement diffusé et repris dans de nombreux travaux connexes. Selon ce formalisme, une théorie logique décrit le comportement normal ou anormal d'un système physique. Ensuite, à partir d'observations mesurées sur le système réel, des hypothèses (appelées diagnostics) peuvent être émises afin d'expliquer son éventuelle mauvais fonctionnement. Ainsi, bon nombre de travaux se ramènent au cadre de la logique propositionnelle (bien qu'il existe des langages plus expressifs) pour décrire le comportement d'un système afin de bénéficier de l'efficacité des algorithmes dans ce domaine. Dans ce cadre, diagnostiquer un système observé se ramène au problème de compilation d'impliquants premiers, et plus généralement au problème de compilation de base de connaissance [4].

L'intérêt des travaux d'IA récents pour un passage à un contexte distribué est devenu plus important ces dernières années, notemment depuis l'ajout de couches sémantiques sur les réseaux fortement distribués [1]. Dans de tels systèmes, que l'on peut apparenter à des systèmes pairs-à-pairs dans lesquels l'état des pairs serait relativement "statique" (l'hypothèse actuelle est d'avoir un réseau suffisamment statique pour mener à bien la requête courante). Dans ces réseaux, tous les pairs exécutent le même algorithme et travaillent à un objectif commun. Ce contexte peut inclure deux types de dispositifs : un pair peut par exemple communiquer avec n'importe quel autre pair où qu'il soit (généralement par l'intermédiaire de table de hachage, ce qui est le cas dans le cadre P2P classique), mais on peut aussi restreindre les communications possible d'un pair à ses voisins directs, comme par exemple les réseaux sociaux, les circuits électroniques et les services Web. Dans ce dernier cadre, le raisonnement est basé sur l'équivalence

logique de variables entre pairs (variables partagées), lesquelles définissent localement la connaissance partagée et le voisinage d'un pair.

Dans cet article, nous nous intéressons plus précisément au problème de diagnostic de systèmes distribués, où chaque pair du système ne connait que ses voisins, ainsi que son modèle logique traduisant un mode de bon ou de mauvais fonctionnement, incluant éventuellement les variables partagées et les observations. Étant donné la connaissance d'un pair, le problème qui nous intéresse ici est de construire des diagnostics pour le système global. Notre approche calcule directement l'ensemble des diagnostics globaux (incluant les minimaux pour l'inclusion ensembliste et pour la cardinalité) sans l'analyse des conflits, un calcul difficile qui est généralement la première étape et le premier goulot d'étranglement de tous les précédents diagnostiqueurs à base de modèles, même si des algorithmes efficaces peuvent être utilisé [15].

Dans le but d'assurer que les diagnostics et les observations soient cohérentes, nous considérons des systèmes distribués "statiques", c'est à dire, intuitivement, que le système reste cohérent tout au long du calcul entre l'ensemble des modèles des pairs connectés au début du calcul jusqu'à la fin. Bien que cette hypothèse ne soit bien entendu pas envisageable pour les systèmes pair à pair classiques, elle demeure réaliste pour les applications distribués comme les services web, les circuits embarqués ou les réseaux sociaux. Dans certains cas particuliers, des couches supplémentaires permettant la mémorisation d'événements passés et des compteurs peuvent simuler l'hypothèse d'un système statique.

Nous rappelons tout d'abord, section suivante, les principes du diagnostic à base de modèles et introduisons quelques notations. Section 3, nous étendons le diagnostic à base de modèles aux Formules Normales Disjonctives. Section 4, nous introduisons les fondements d'un raisonnement distribué pour le diagnostic. Enfin, section 5, nous présentons l'algorithme distribué puis nous terminons par les travaux liés et la conclusion.

**Exemple 1** Nous illustrons l'approche par un exemple simplifié de paiement en trois fois par internet voir Figure 1. Le service de validation de commande (OVS), demande au service d'e-shopping (ES) un accord d'achat à crédit (hpPurch). Dans le but de maximiser ses opportunités de vente ES attend l'accord de la banque du client (bkAprvl) ou encore celui d'une agence de crédit (laAprvl). Le service d'achat à crédit de la banque (HPS) et le service de l'agence de crédit (LAS) vérifient tous deux la validité de la carte bancaire du client (valCC) par un appel au service des cartes de crédit (CCS) de l'acheteur. Dans un premier temps nous réduisons notre analyse au système (HPS) et nous nous référons à sa description qui peut être vue comme la conjonction des sous systèmes de validation de transaction (TA), vérification de solvabilité (SC) vérification d'options (OC).

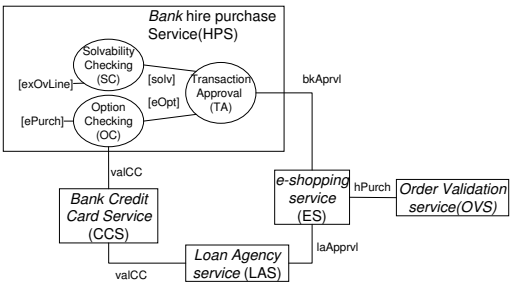


FIG. 1 – Validation d'un paiement en trois fois par internet

## 2 D'un diagnostic en FNC à un diagnostic en FND

Nous supposons le lecteur familiarisé avec les notations usuelles de logique propositionnelle. Un littéral est une variable $v$ ou sa négation $\neg v$. Étant donné un ensemble $L$ de littéraux, on note par $\overline{L}$ l'ensemble de ses littéraux opposés. Une Forme Normale Conjonctive (FNC) est une conjonction de disjonctions de littéraux. Une Forme Normale Disjonctive FND est une disjonction de conjonctions de littéraux. Pour simplifier, nous identifierons une telle formule par son ensemble d'ensembles de littéraux. Nous notons $T^{\wedge}$ (resp. $T^{\vee}$) l'ensemble d'ensembles de littéraux correspondant à une FNC (resp. FND).

Un *modèle* est un ensemble de littéraux qui, lorsqu'il est assigné à vrai, permet d'évaluer la formule à vrai. Une formule $f$ est satisfiable ou consistante, notée par $f \not\models \bot$, s'il existe au moins un modèle pour $f$. Soient $f_1$ et $f_2$ deux formules, si tout modèle de $f_1$ est aussi modèle de $f_2$, ce qui se note $f_1 \models f_2$, alors $f_1$ est appelé *impliquant* de $f_2$ et $f_2$ est appelé *impliqué* de $f_1$. Les clauses minimales (vis-à-vis de l'inclusion ensembliste) impliquées par une formule sont appelées les *impliqués premiers*. De même, les produits minimaux impliquant une formule sont appelés *impliquants premiers*. On peut noter que l'ensemble des impliqués premiers sont exprimés en FNC alors que l'ensemble des impliquants premiers s'expriment en FND. Étant donnée une formule $f$ et un sous ensemble $V$ de ses variables, la restriction de $f$ sur $V$ est notée par $f|_{\{V\}}$ et correspond à appliquer récursivement sur $f$ l'opération de décomposition de Shannon sur toutes les variables $x$ de $f$ qui n'apparaissent pas dans $V$. Cette opération est connue sous le nom de « forgetting » en compilation de bases de connaissances, et est bien connue pour être un problème NP-Difficile. Cependant, quand $f$ est sous forme FND, l'opération de restriction est réduite à la restriction syntaxique sur le vocabulaire des produits de $f$. La *restriction* de $T^{\vee}$ sur un ensemble de littéraux $L$ se définie par $T^{\vee}|_{\{L\}}$ = $\{I^L \mid \exists I \in T^{\vee}$ t.q. $I^L = I \cap L\}$, ce qui n'est plus un problème difficile.

## 2.1 Principe du diagnostic à base de modèles

Nos travaux sont basés sur le diagnostic à base de modèles de [8, 7], appliqué au cadre propositionnel. Initialement, un système observé est défini comme un triplet (SD,COMPS,OBS) où SD est une formule de la logique du 1er ordre décrivant le comportement du système, OBS est une formule décrivant les observations (revenant dans la plupart des cas à une assignation de valeurs aux variables observables) et COMPS est l'ensemble des composants surveillés. Chacun de ces composants apparaît en tant qu'argument du prédicat prédéfini $Ab()$ dans SD (ex : $Ab(C_i)$ signifie que $C_i$ est anormal). En logique propositionnelle, nous réunissons ces ensembles dans une seule théorie, $T$, avec la convention de nommage suivante : une variable $okC_i$ encode un mode de comportement correct pour le composant $C_i$, c-à-d $\neg Ab(C_i)$. On appelle $F$ l'ensemble des littéraux de modes négatifs {...,$\neg okC_i$,...} représentant des composants défaillants. Pour un observable (booléen) encodé par une variable $v$, l'observation élémentaire $v = a$ est traduite par $v$ si $a$ est égal à 1 et $\neg v$ si $a$ est égal à 0.

**Exemple 2 (Modéliser le système)** Un fonctionnement correct du service TA ($okTa$) approuvera un achat à crédit (bkAprvl) si le client est solvable ($solv$) et s'il satisfait la condition ($eOpt$) de OC. La règle pour TA peut se réécrire par : $f(TA)$ : $okTA \Rightarrow (solv \wedge eOpt \Leftrightarrow bkAprvl)$. Un comportement normal de SC ($okSC$) considérera un client solvable si il ne dépasse pas son découvert autorisé ($\neg exOvLine$). Nous obtenons : $f(SC)$ : $okSC \Rightarrow (\neg exOvLine \Leftrightarrow solv)$. Un fonctionnement normal de OC ($okOC$) satisfera ($eOpt$) si le client a demandé la possibilité d'effectuer des achats à credit par internet ($ePurch$) et si sa carte de crédit est valide ($valCC$). Il y a que deux possibilités de fautes pour $OC$ : soit $ePurch$ conserve sa valeur par défaut alors que le client a demandé l'option, ou encore la carte de crédit du client est reconnue comme non valide alors qu'elle l'est. Le système de vérification d'options peut donc s'encoder par : $f(OC) : okOC \Rightarrow (ePurch \wedge valCC \Leftrightarrow eOpt) \wedge (\neg okOC \Rightarrow \neg valCC \vee \neg ePurch)$. Le comportement de HPS est la conjonction : $f(HPS) : f(OC) \wedge f(SC) \wedge f(TA)$.

Le théorème 3 de [7] montre que les diagnostics minimaux sont des impliquants premiers de la conjonction des conflits minimaux où les conflits minimaux (appelés ensemble de conflits minimaux [13] ou ensemble de conflits minimaux positifs dans [7]) sont les impliquants premiers de la formule $SD \wedge OBS$ restreints aux littéraux de modes de $F$. Intuitivement, un conflit minimal fait référence à un ensemble de composants dont au moins un est défaillant. Les diagnostics minimaux sont par conséquent les plus petites conjonctions de composants défaillants expliquant toutes les fautes, étant données des observations.

**Définition 1 (Diagnostics minimaux)** *Soit $T$ une théorie décrivant un système observé et $F$ un ensemble consistant de littéraux de mode négatifs.*

$$\triangle \subseteq F \text{ est un diagnostic ssi } T \cup \triangle \cup \overline{\{F \setminus \triangle\}} \not\models \bot$$

*On note $Diag(T)$ l'ensemble des diagnostics de $T$ et $min_{\subseteq}(Diag(T))$ son ensemble de diagnostics minimaux.*

Intuitivement, cette définition statut qu'étant donné n'importe quel diagnostic minimal $\triangle$ d'une erreur observée, on peut supposer que tous les composants n'apparaissant pas dans $\triangle$ sont corrects. Nous remarquons aussi que, parce qu'on peut restreindre l'ensemble des erreurs possibles ($(\neg okOC \Rightarrow \neg valCC \vee ePurch)$ dans le précédent exemple, un diagnostic peut ne pas être étendu en supposant tous les composants $C'$ incorrectes (deux littéraux de mode étendant un diagnostic peuvent être incohérents).

**Exemple 3 (Conflits et diagnostic : scenario 1)** Supposons le scénario suivant : la banque a approuvé un achat à crédit pour une transaction alors que le client avait dépassé son découvert autorisé. Il avait une carte de crédit valide mais il avait demandé à ce que la banque n'autorise pas les achats par internet. Dans ce scenario, le service de la banque ne suit pas son comportement attendu. Par conséquent nous recherchons le sous ensemble minimal de { TA, OC, SC } qui est fautif et qui est consistant à la fois avec la formule $f(HPS)$ et les observations {$exOvLine$, $valCC$, $\neg ePurch$, $bkAprvl$ }. Ici les conflits minimaux sont : $(\neg okSC \vee \neg okTA)$ et $(\neg okOC \vee fV \vee \neg okTA)$. Les diagnostics minimaux satisfaisant les conflits sont : $(\neg okTA)$ et $(\neg okSC \wedge \neg okOC)$.

La plupart des travaux sur le diagnostic à base de modèles calculent en premier lieu l'ensemble des conflits se restreignant aux littéraux de mode. C'est seulement après l'accomplissement de cette première étape que sont calculés les diagnostics minimaux. Cependant, cette méthode n'a que peu d'espoirs d'aboutir lors d'un diagnostic en ligne, dans la mesure où tous les conflits doivent être connus avant que le premier diagnostic ne puisse être retourné. Le calcul des conflits a été initialement justifié par le fait que, dans le monde réel, la description logique des systèmes est supposée proche des FNC. Si besoin, de nouvelles variables sont ajoutés afin de contenir l'éventuelle explosion combinatoire lors de la transcription en FNC. Cependant, une représentation en FND d'un circuit peut aussi être d'un grand intérêt, si on s'assure que l'ensemble $F$ des variables de mode est consistant, ce qui signifie qu'aucune variable apparaît à la fois positivement et négativement dans $F$, alors, si $T^{\vee}$ est la description d'un system observé, chaque produit de la restriction $T|_{\{F\}}$ est un diagnostic, non nécessairement minimal.

**Lemme 1** *Soit $T^{\vee}$ la description d'un système observé sous forme FND et $F$ un ensemble consistant de littéraux de mode, alors*

$$\forall I \in T^{\vee}, I|_{\{F\}} \in Diag(T)$$

**Preuve** Pour tous $I \in T^{\vee}$, $I$ est un impliquant de $T$ donc est trivialement consistant avec $T$. Considérons l'ensemble $\{F \setminus$

$I|_{\{F\}}\}$, qui ne contient aucun littéral de $I|_{\{F\}}$, on a $T \cup I|_{\{F\}} \cup \{F \setminus I|_{\{F\}}\}$ qui est consistant avec $T$ et par définition $I|_{\{F\}}$ est un diagnostic

Par conséquent, si on calcule au moins un impliquant de $T$, on obtient au moins un diagnostic sans pour autant attendre l'ensemble des conflits. En pratique nos suppositions sur la représentation FND, qui semble de prime abord forte, de $T$ peut être modérée par le fait que les implicants peuvent être calculés de façon incrémentale par un Solver SAT ou encore déduit par une représentation compacte et compilée de T qui permettrait une production efficace de modèles. Cependant, pour de petits système ou de larges systèmes mais distribués, la transformation d'une CNF vers une FND peut certainement être effectué. Le théorème suivant permet d'affirmer que les diagnostics minimaux sont contenus dans n'importe qu'elle FND encodant le système observé.

**Théorème 1** *Soit $T$ une description d'un système observé, et $F$ un ensemble consistant de littéraux de mode :*

$$min_{\subseteq}(Diag(T)) = min_{\subseteq}(T^{\vee}|_{\{F\}})$$

**Idée de la preuve** Soit $\triangle$ un diagnostic minimal, par la définition 1, $\triangle \cup \overline{\{F \setminus \triangle\}}$ est consistant avec le système observé. De plus, pour toutes FND encodant le système, il existe un impliquant $I$ consistant avec $\triangle \cup \overline{\{F \setminus \triangle\}}$. Puisque $I$ est consistant avec $\triangle \cup \overline{\{F \setminus \triangle\}}$ on a $I|_{\{F\setminus\triangle\}} = \emptyset$ et par conséquent $I|_{\{F\}} = I|_{\{\triangle\}}$. Parce que l'on sait, par le lemme 1 que $I|_{\{F\}}$ est un diagnostic on a $: I|_{\{\triangle\}} = \triangle$.

**Exemple 4 (Extraire les diagnostic d'une FND)**
Soit $F_{HBS}$ = {$\neg okTA$,$\neg okOC$,$\neg okSC$}. Les littéraux de mode négatifs et $T^{\vee}_{HBS}$ la formule FND encodant la description du système et les observations.

$$T^{\vee}_{HBS} = \begin{array}{l}(\neg okTA \wedge \neg okSC \wedge \neg eOpt \wedge okOC)\vee \\ (\neg okTA \wedge \neg okSC \wedge \neg okOC)\vee \\ (\neg okTA \wedge \neg solv \wedge \neg okOC)\vee \\ (\neg okTA \wedge \neg okSC \wedge \neg eOpt)\vee \\ (\neg okTA \wedge \neg solv \wedge \neg eOpt)\vee \\ (\neg okSC \wedge \neg okOC \wedge eOpt \wedge solv)\end{array}$$

Pour simplifier, nous avons omis la conjonction de littéraux observés $exOvLine \wedge valCC \wedge \neg ePurch \wedge kAprvl$. Finalement, après restriction sur $F_{HBS}$ et l'élimination par subsomption, on obtient deux diagnostics $\{\neg okTA, (\neg okSC \wedge \neg okOC)\}$.

# 3 Diagnostic de systèmes distribués

Nous formalisons notre problème de diagnostic distribué en nous basant en grande partie sur notion de système d'inférence pair-à-pair (P2PIS) proposé par [9] et étendu dans [1] pour le raisonnement distribué. Dans un P2PIS, un pair d'inférence a une connaissance partielle du système global (généralement restreint à son voisinage) et peut modéliser une connaissance (locale) connue de lui seul. Dans notre approche, un pair d'inférence pourra par exemple modéliser le comportement d'un pair du système réel, un service web, ou encore un sous circuit d'un système distribué. Notons par $T$ la description de la théorie globale observée. $T$ peut être vue comme la conjonction de toutes les théories locales $T_p$ des pairs $p$. Dans notre approche, $T$ ne devra jamais être explicité clairement et une contrainte de confidentialité devra être assurée sur les connaissances locales. $T$ se construit à partir du vocabulaire global suivant : $V$ se partitionnant en un ensemble des variables partagées $Sh$ et un ensemble des variables locales $Loc$.

- $Sh = \{v | \exists p, p'$t.q. $v$ apparaît dans $T_p$ et dans $T_{p'}\}$
- $Loc = \{v | \nexists p$t.q. $v$ apparaît uniquement dans $T_p\}$

En plus de ces partitions, nous devons ajouter les variables de mode dans le but de pouvoir diagnostiquer le système. On dénote par $F$ l'ensemble de toutes les variables de mode du système. Une variable de $F$ ne peut rester locale à un pair, autrement aucun diagnostic global n'est possible. Ainsi, le réseau permet le passage de formules construites à partir des variables : $Sh \cup F$. Nous notons par $V_p, Sh_p, Loc_p, F_p$, le vocabulaire, le vocabulaire partagé, le vocabulaire local et les symboles de mode d'un pair d'inférence $p$ respectivement.

## 3.1 Un réseau de FND

Nous avons supposé, au cours de la section précédente, que nous pouvons travailler directement sur la transcription FND de $T$. Dans la mesure où $T$ est ici une conjonction de formules, nous supposerons que la transcription en FND est réalisable pour chaque théorie $T_p$, qui doivent être en pratique de taille raisonnable. Si l'hypothèse de la transcription FND de la théorie globale peut être considérée comme non réaliste, des pairs de petite taille, eux, peuvent admettre une petite théorie FND. Si tous les $Tp$ sont en FND, alors écrire $T$ en FND peut s'effectuer par la propriété de distribution de $\wedge$ sur $\vee$. Plus formellement, nous utilisons l'opérateur suivant à cet effet :

**Définition 2 (Distribution ($\otimes$))**
$T^{\vee}_1 \otimes T^{\vee}_2 = \{I_1 \wedge I_2 | I_1 \in T^{\vee}_1, I_2 \in T^{\vee}_2, I_1 \wedge I_2 \not\models \bot\}$

Nous pouvons dans un premier temps constater que les produits inconsistants sont éliminés. Si le résultat est minimisé, cet opérateur est exactement l'opérateur de *clause-distribution* de [15], mais appliqué aux FND et aux produits.

De manière à prendre en compte la confidentialité, mais aussi pour des raisons d'efficacité, nous pouvons introduire le lemme suivant, qui stipule qu'au lieu de distribuer toutes les théories en premier lieu, puis restreindre le résultat aux variables de mode, on peut tout d'abord restreindre toutes les théories aux variables partagées et aux variables de mode, sans perte de généralité du résultat final.

**Lemme 2** *Soit $T^{\vee}$ la description d'un système distribué observé et $F$ un ensemble consistant de littéraux de mode :*

$$(\otimes T^{\vee}_p)|_{\{Sh,F\}} = \otimes(T^{\vee}_p|_{\{Sh_p,F_p\}})$$

**Idée de la preuve** Soit $I$ (resp. $I'$) un impliquant de $T_p^\vee$, (resp. $T_{p'}^\vee$), les symboles locaux à $I$ n'apparaissent pas dans $I'$, Par conséquence les inconsistances entre $I$ et $I'$ ne peuvent provenir que des variables partagées.

Avec ce lemme et le théorème 1, on peut montrer que les diagnostics minimaux peuvent être calculés à partir des variables partagées et des littéraux de modes.

**Théorème 2** *Soit $T^\vee$ la description d'un système observé global, $F$ un ensemble consistant de littéraux de mode :*

$$min_\subseteq(Diag(T)) = min_\subseteq((\otimes(T_p|_{\{Sh_p,F_p\}}))|_{\{F\}})$$

**Idée de la preuve** Soit $T^\vee$ la formule FND encodant le système observé global. Il est equivalent à $\otimes T_p^\vee$. Par le théorème 1, nous savons que $T^\vee$ contient les diagnostics minimaux, et par le lemme 2 on peut remplacer $\otimes T_p^\vee$ par son expression sur les variables partagées et les littéraux de mode $\otimes(T_p|_{\{Sh_p,F_p\}})$.

### 3.2 Composition des diagnostics par un arbre

Dans cette partie, nous nous focalisons sur la distribution de diagnostics consistants entre pairs. Nous avons décidé d'étendre la solution de [1] et pouvons considérer que, lorsqu'un pair le décide, il peut initier le calcul de diagnostics en demandant de l'aide à ses voisins. Lorsqu'un pair reçoit une demande de diagnostic d'un initiateur, il propage la requête à ses voisins si besoin et répond dès que possible à l'initiateur en tenant compte de ses observations et de son modèle de fonctionnement. Par conséquent, la requête de diagnostic inondera le réseau de haut en bas (en considérant l'arbre couvrant généré pour l'occasion), en partant du pair initial. Les réponses convergeront vers le pair initiateur en traversant le réseau de bas en haut.

Le calcul implicite (puisque totalement distribué) de l'arbre couvrant a une importance cruciale pour calculer efficacement la distribution entre les théories des pairs. Notons $A_p$ le sous arbre de racine $p$ et $child(A_p, A_{p'})$ une relation entre $A_{p'}$ et $A_p$ tel que $A_{p'}$ est un sous arbre de $A_p$. Nous notons $Loc_{A_p}$ l'ensemble des littéraux apparaissant uniquement dans le vocabulaire des pairs du sous arbre $A_p$ et $Sh_{A_p}$ le vocabulaire partagé entre $A_p$ et d'autres pairs dans le système. $T^{A_p}$ représente la théorie du sous arbre enraciné en $p$.

$$T^{A_p} = \begin{cases} T^\vee|_{\{F_p,Sh_p\}}, \text{ si } \nexists p', child(A_p, A_{p'}) \\ (T^\vee|_{\{F_p,Sh_p\}} \otimes T^{A_{p'}})|_{\{Sh_{A_p},F_{A_p}\}}, \text{ sinon} \end{cases}$$

Le théorème suivant montre que nous pouvons calculer l'ensemble des diagnostics globaux en supprimant au fur et à mesure les variables partagées correspondant à une connaissance locale d'un sous arbre.

**Théorème 3** *Soit $T$ la description du système observé global, $Child(A_p, A_{p'})$ une relation définissant un arbre sur $T$ racine r. alors :*

$$min_\subseteq(Diag(T)) = min_\subseteq(T^{A_r})$$

**Idée de la preuve** Nous utilisons le 2 et prouvons inductivement que $\forall p, T^{A_p} = (\underset{q\in A_p}{\otimes} T_q^\vee|_{\{Sh_q,F_q\}})|_{\{Sh_{A_p},F_{A_p}\}}$. Concernant la racine $r$, nous notons que $Sh_{A_r}$=$\emptyset$, par consequence $min_\subseteq(T^{A_r})$ contient uniquement les diagnostics minimaux.

Ainsi, intuitivement, dès que nous savons qu'une variable n'impliquera pas une inconsistance autre part dans l'arbre, nous la supprimons. Comme les réponses remontent à la racine, les pairs filtrent les variables inutiles et ainsi nous espérons réduire le nombre et la taille des réponses possibles.

## 4 Algorithme

Nous présentons maintenant notre algorithme de calcul de diagnostics par des arbres distribués : M2DT. Les messages sont soit des requêtes $reqDiag$ pour initier le diagnostic, soit des réponses $respDiag$ pour renvoyer les diagnostics au pair *parent* (selon la convention de nommage dans les arbres) ou encore une notification de terminaison : $terminate$. Les parents sont uniques et locaux à chaque pair mais peuvent être différents à chaque nouvelle requête de diagnostic global. Notre algorithme se compose des algorithmes :1, 2 et 3. Un pair particulier (ayant $starter_p$ à vrai) débute le calcul et commence par diffuser $reqDiag$ à tous ses voisins.

Dans la suite, nous supposons qu'aucun message n'est perdu, et le graphe des connaissances est connexe. Même si on ne suppose pas un comportement FIFO de tous les messages dans le réseau, nous supposons que le message $terminate$ sera traité après tous les messages $respDiag$ envoyés précédemment sur le même canal de communication. Ceci peut être aisément simulé en estampillant tous les messages dans un réseau non FIFO.

### 4.1 Description de l'algorithme

Pour une requête donnée, chaque pair $p$ maintient un ensemble $waitChild$ de ses voisins pour lesquelles des réponses aux requêtes sont attendues. Initialement, cet ensemble est constitué de tous les voisins. Les réponses sont des impliquants restreints aux variables partagées et aux symboles de mode. Afin de simplifier les notations nous appelons aussi impliquant, un impliquant restreint sur les variables partagées et les littéraux de modes. Pour chacun de ses voisins $p'$, chaque pair $p$ maintient aussi un tableau $TChild$ qui associe un pair $p'$ à une théorie FND $TChild[p']$. Tous les éléments de ce tableau pourront conserver l'ensemble des impliquants (possiblement minimisé) pour lesquels $p$ a déjà eu précédemment au moins une réponse. Initialement, $TChild$ est un tableau vide.

Dans le but de réutiliser le théorème 3, qui permet de supprimer des variables partagées dès que l'on est sûr qu'elles apparaissent uniquement dans le sous arbre courant ($p$ étant considéré comme sa racine), l'algorithme

maintient l'ensemble $Desc_p$ pour tous les pairs $p$. Cet ensemble, qui est initialement vide, conserve les descendants connus de $p$. $Desc_p$ est utilisé dans la fonction $removeSharedVarInSubtree(T^\vee)$ dans le but de retirer toutes les variables partagées inutiles comme nous l'indique le théorème 3. Pour être inutiles, ces variables doivent avoir tous leurs pairs associés dans $Desc_p$. En pratique, toutes les variables partagées sont associées à l'ensemble des pairs qui les utilisent. Avec cette convention, $removeSharedVarInSubtree(T^\vee)$ peut facilement identifier les variables inutiles.

Dans le premier algorithme, ligne 1-2 on peut supprimer $p'$ de $waitChild$ et de $waitTermination$ à cause de la propriété suivante :

**Propriété 1 (exclusion mutuelle de $reqDiag$ et $respDiag$)** *Pour tous pairs p, et tous ses voisins p',p ne peut recevoir qu'un unique $reqDiag$ de $p'$ ou un ensemble de $respDiag$ de $p'$, mais pas les deux.*

Si $p$ reçoit un message $reqDiag$ de $p'$, cela signifie que $p'$ a déjà choisi un autre pair comme parent ($p'$ était à la ligne 6 de l'algo 1) et propose à $p$ d'être un de ses fils. Par conséquent, $p$ n'a pas à attendre de réponse de la part de $p'$ (ligne 13 de alg. 1 et ligne 12 de alg. 3) les réponses sont seulement envoyées au pères). Dans tous les cas, le premier message reçu par $p$ est un $reqDiag$ (p ne peut recevoir aucun message $respDiag$ avant, pour la bonne raison qu'aucun pair n'a pu recevoir un message $reqDiag$ de $p$ et par conséquent aucun n'a pu choisir $p$ comme parent). Lignes 4-7, $p$ choisit l'expéditeur de son premier message reçu comme parent et diffuse la requête à ses voisins. Nous pouvons aussi noter que, si nous restreignons les algorithmes à l'algorithme 1, lignes 1-7, alors il est facile de montrer la propriété suivante :

**Propriété 2 (M2DT construit un arbre couvrant)** *Tous les pairs reçoivent au moins un reqDiag d'un de ses voisins. À la fin, la relation (parent,p), qui est distribuée à travers les pairs, définit un arbre couvrant distribué à travers le graphe des connaissances.*

Les lignes 8-9 de l'alg. 1 et lignes 4-16 de l'alg.2 ont le même but et sont très proches. Dans l'alg. 1, $p$ satisfera le test de la ligne 8 si $p$ est une feuille de l'arbre couvrant, ou s'il a reçu un message $reqDiag$ de son dernier voisin attendu. Dans les deux cas il doit envoyer les impliquants ( $Tmp^\wedge$) de son sous-arbre à son père, avec les restrictions usuelles sur le vocabulaire. Si $p$ est une feuille, alors $waitTermination$ est vide (une feuille reçoit uniquement des messages $reqDiag$), par conséquent $p$ envoie un message de terminaison à son père. Si $p$ n'est pas une feuille il peut encore recevoir des impliquants de ses fils, ce qui est pris en compte dans alg. 2. Avant de décrire le second algorithme, examinons la propriété suivante :

**Algorithm 1** $p$ receives msg(reqDiag) from $p'$

1: $waitChild \leftarrow waitChild \setminus p'$
2: $waitTermination \leftarrow waitTermination \setminus p'$
3: **if** $starter_p$ **then** return
4: **if** $parent_p$ is not set **then**
5: $parent_p \leftarrow p'$
6: send msg(reqDiag) to all $p''$ of $waitChild$
7: **end if**
8: **if** $waitChild = \emptyset$ **then**
9: $Tmp^\vee \leftarrow (\otimes TChild) \otimes T_p^\vee|_{\{Sh_p,F_p\}}$
10: **if** $Tmp^\vee$ is consistent **then**
11: $removeSharedVariablesInSubTree(Tmp^\vee)$
12: **for all** $I \in Tmp^\vee$ **do**
13: send msg(rspDiag,$I$,$Desc_p \cup p$) to $parent_p$
14: **end for**
15: **if** $waitTermination = \emptyset$ **then**
16: send msg(terminate) to $parent_p$
17: **end if**
18: **end if**
19: **end if**

**Propriété 3 (waitChild sera vide)** *Pour tous pair $p$ et pour toute requête initiale, il existe un moment à partir duquel $waitChild$ sera vide.*

**Idée de la preuve** Nous avons vu que tous les pairs exécutent une seule fois la ligne 6 de l'alg. 1, avec $waitChild$ égal aux voisins moins le parent. Soit $v_{p,p'}$ une variable partagée entre $p$ et $p'$. Nous sommes certain que soit $p$ a envoyé $reqDiag$ à $p'$ ou que $p'$ a envoyé $reqDiag$ à $p$. Considérons, sans perte de généralité, le premier cas. L'ensemble $waitChild$ de $p'$ ne contient plus $p$. Supposons que $p$ ne retirera jamais $p'$ de son ensemble $waitChild$. Par conséquent, $p'$ n'a jamais envoyé ni de $reqDiag$ ni de $respDiag$ à $p$. Si $p'$ avait choisit $p$ comme père, nous aboutissons directement à une contradiction ($p'$ a dû envoyer un $respDiag$ à $p$) ; si $p'$ n'a pas choisi $p$ comme père, alors nous avons aussi une contradiction ($p'$ a du un envoyé un $reqDiag$ à p).

Dans l'algorithme 2, ligne 5, $I_{p'}$ est le dernier impliquant reçu du sous arbre de racine $p'$. Soit $Tmp^\vee = T_p^\vee|_{\{Sh_p,F_p\}} \otimes \bigotimes_{p" \neq p'} TChild[p"]$ . $p$ essaie de prolonger $I_{p'}$ par tous les impliquants de $Tmp^\vee$ possibles. Tous ces nouveaux impliquants prolongés sont aussi des impliquants du sous-arbre de racine $p$ et doivent être envoyés au père de $p$. Parce que waitChild est vide, l'ensemble des fils ne peut plus évoluer, mais tous les fils peuvent continuer à envoyer de nouveaux impliquants à prolonger.

Lorsque le pair initiateur reçoit la notification de terminaison de ses fils, sa variable $Result^\vee$ contient l'ensemble des diagnostics minimaux de tout le système.

**Exemple 5 (illustration de M2DT scénario 2)** Nous considérons ce scénario, illustré Figure 2 : un client effectue un achat à crédit par internet alors que sa carte de crédit n'est pas

**Algorithm 2** $p$ receives msg(respDiag, $I_{p'}$,$Desc_{p'}$) from $p'$

1: $waitChild \leftarrow waitChild \setminus p'$
2: $Desc_p \leftarrow Desc_p \cup Desc_{p'}$
3: $addOrCreateWith(I_{p'}, TChild, p')$
4: **if** $waitChild = \emptyset$ **then**
5: $Prol^{\vee}_{I_{p'}} \leftarrow T^{\vee}_p|_{\{Sh_p,F_p\}} \otimes I_{p'} \underset{p" \neq p'}{\otimes} TChild[p"]$
6: **if** $Prol^{\vee}_{I_{p'}}$ is consistent **then**
7: $removeSharedVariablesInSubTree(Prol_{I_{p'}})$
8: **if** $starter_p$ **then**
9: $Tresult^{\vee} \leftarrow min_{\subseteq}(Tresult^{\vee} \cup Prol_{I_{p'}})$
10: **else**
11: **for all** $I_p \in Prol^{\vee}_{I_{p'}}$ **do**
12: send (respDiag,$I_p$,$Desc_p \cup p$) to $parent$
13: **end for**
14: **end if**
15: **end if**
16: **end if**

**Algorithm 3** $p$ receives msg(terminate) from $p'$

1: $waitTermination \leftarrow waitTermination \setminus p'$
2: **if** $waitTermination = \emptyset$ **then**
3: send msg(terminate) to $parent$
4: **end if**

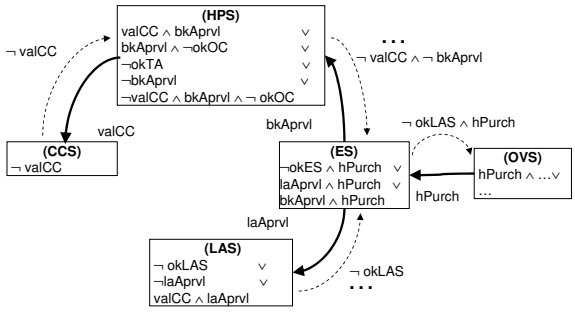


FIG. 2 – M2DT algorithm

valide. Les services $CCS$ et $ES$ observent $\neg valCC$ et $hPurch$. $OVS$ débute l'analyse et envoie une requête de diagnostic à $ES$. $ES$ commence le calcul de $T^{\vee}_{ES}|_{\{Sh_{ES},F_{ES}\}}$ et diffuse la requête à ses voisins. Simultanément $HPS$ et $LAS$ reçoivent la requête de diagnostic de $ES$. Ils font les mêmes opérations que leur initiateur et diffusent à leur tour la requête à $CCS$. $CCS$ reçoit en premier lieu la requête de diagnostic de $HPS$, il calcule $T^{\vee}_{CCS}|_{\{Sh_{CCS},F_{CCS}\}}$ et retransmet la requête à $LAS$. Lorsque $LAS$ reçoit la requête de diagnostic de $CCS$, il a déjà reçu une requête de $ES$. Par conséquent $LAS$ ne répond pas à $CCS$. A cette étape $LAS$ a reçu un message de tous ses voisins, il commence à envoyer ses impliquants à son parent : $ES$. Parallèlement, lorsque $CCS$ reçoit la requête de $LAS$, il n'y répond pas et envoie son impliquant $\neg valCC$ à $HPS$. $ES$ reçoit $\neg okLAS$ de $LAS$ et $HPS$ reçoit $\neg valCC$ de $CCS$. A cet étape, $ES$ n'a pas encore reçu de message de $HPS$. $HPS$ envoie à $ES$ un nouvel impliquant construit à partir de $\neg bkAprvl \wedge \neg valCC$ extrait de sa théorie et de $\neg valCC$ reçu de $CCS$. $ES$ reçoit l'impliquant de $HPS$ et construit $\neg bkAprvl \wedge hPurch \wedge \neg valCC$ à l'aide de ses impliquants et des impliquants reçus. $ES$ retire $\neg bkAprvl$ et $\neg valCC$ qui sont des variables partagées présentent uniquement dans le sous arbre de racine $ES$ et envoie $\neg okLAS \wedge hPurch$ à $OVS$. A cet instant $OVS$ a son premier diagnostic : $\neg okLAS$.

## 5 Travaux connexes

Bien entendu, notre approche a été précédée par de nombreux travaux connexes. On trouve ainsi dans [11] et [2] des extensions des principes des ATMS [6] pour le calcul incrémental d'ensembles de conflits dans un système distribué. Cependant, ces méthodes ont toutes besoin d'attendre d'avoir l'ensemble de tous les conflits avant de pouvoir renvoyer le premier diagnostic. D'autre travaux tentent aussi de profiter de la topologie du système, par exemple en utilisant les propriétés de décomposition du système [5, 12]. De même, [10] cherche un ensemble de diagnostics dans un graphe partitionné en affectant certaines valeurs pour les variables partagées, tout en maintenant la consistance locale de chaque pair. Le diagnostic global est alors distribué implicitement dans tous les diagnostics locaux. Cette méthode ne permet cependant pas de garantir la minimalité du résultat et suppose, de plus, une description globale du système, ce qui n'est pas notre cas. Enfin, certains travaux [3] tentent de synchroniser les diagnostics locaux de chacun des " agents " (pairs), de manière à ce qu'ils ne contiennent qu'une représentation compacte du diagnostic global à la fin du calcul. L'algorithme cherche ainsi l'ensemble des diagnostics minimaux de cardinalité minimale alors que nous cherchons ici l'ensemble de tous les diagnostics minimaux. Dans [14], les agents mettent à jour leurs ensembles de diagnostics de manière à garantir la consistance avec les diagnostics globaux. Cependant, aucune garantie n'est donnée sur le fait que la combinaison des diagnostics locaux des agents puisse donner un diagnostic global minimal.

## 6 Conclusion

Dans cet article, nous avons présenté un algorithme distribué permettant de calculer les diagnostics minimaux d'un système distribué. Grâce au calcul préalable d'un arbre couvrant - mais distribué et calculé en même temps que la requête - du système, basé sur le graphe d'acquaintance des pairs, notre algorithme permet un calcul en "Anytime" des diagnostics. De plus, nous nous appuyons sur une représentation sous forme normale disjonctive des théories locales des pairs, ce qui permet de calculer directement les diagnostics intéressants sans passer par une génération

préalable des conflits. On doit cependant bien noter ici que les pairs n'ont pas besoin d'écrire explicitement leurs théories en FND. En effet, cela peut très bien se faire à la volée, ce qui permettrait à notre algorithme d'être déployé sur des théories sous CNF, ou compilées dans un langage compact permettant d'énumérer les modèles de manière simple. Le second avantage de notre approche réside sur la topologie du réseau, qui est clairement décomposable suivant les variables partagées entre les pairs, qui devraient être en faible nombre par rapport à l'ensemble global des variables du système. De plus, nous pouvons profiter de la puissance de calcul distribué de tout le système, en répartissant l'effort de calcul sur un grand nombre de pairs. Enfin, en restreignant le vocabulaire intéressant dès que possible lors du calcul, nous espérons pouvoir observer des gains pratiques importants. Nous avons implanté M2DT et avons lancé quelques exemples jouets. Nos premières observations, reportées ici de manière purement informelle et illustrative, tendent à montrer que la racine n'a effectivement pas à supporter tout le calcul des diagnostics, même si elle doit bien les stocker tous. Des noeuds intermédiaires effacent en effet les variables partagées uniquement dans leur sous-arbre, ce qui permet de simplifier grandement le calcul des noeuds parents de l'arbre couvrant considéré. Nous sommes actuellement sur la voie d'un déploiement de M2DT sur des exemples plus proches du monde réel de manière à mesurer en pratique ses performances.

## Références


[1] Philippe Adjiman, Phillipe Chatalic, M.-C Rousset, and Laurent Simon. Distributed reasoning in a peer-to-peer setting. *IJCAI'2005*, 2005.

[2] C. Beckstein, R. Fuhge, and G. Kraetzschmar. Supporting assumption-based reasoning in a distributed environment. *12th International Workshop on Distributed Artificial Intelligence*, 1993.

[3] Jonas Biteus, Erik Frisk, and Mattias Nyberg. Distributed diagnosis by using a condensed local representation of the global diagnoses with minimal cardinality. *International Workshop on Principles of Diagnosis (DX-06), Spain*, 2006.

[4] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of AI Research*, 17 :229–264., 2002.

[5] Adnan Darwiche. Model-based diagnosis using structured system descriptions. *Journal of Artificial Intelligence Research*, 8 :165–222, 1998.

[6] Johan de Kleer. An assumption-based tms. *Artificial Intelligence*, 28 :127–162, March 1986.

[7] Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing diagnoses and sytems. *Artificial Intelligence*, 56 :197–222, August 1992.

[8] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1) :97–130, April 1987.

[9] A. Halevy, Z. Ives, and I. Tatarinov. Schema mediation in peer data management systems. *In : ICDE'O3*, pages 505–516, March 2003.

[10] J. Kurien, X. Koutsoukos, and F. Zhao. Distributed diagnosis of networked, embedded systems. *International Workshop on Principles of Diagnosis (DX-02), Austria*, 2002.

[11] Cindy L. Masson and Rowland R. Johnson. Datms : A framework for distributed assumption based reasonning. In *Distributed artificial intelligence*, volume 2, pages 293–317. Morgan Kaufman Publishers Inc., 1989.

[12] Gregory Provan. A model-based diagnosis framework for distributed systems. *International Workshop on Principles of Diagnosis (DX-02), Austria*, 2002.

[13] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1) :57–96, April 1987.

[14] Nico Roos, Annette ten Teije, and Cees Witteveen. A protocol for multi agent diagnosis with spatially distributed knowledge. *2nd International Joint Conference on Autonomous Agents and Multiagent Systems, Australia*, 2003.

[15] L. Simon and A. del Val. Efficient consequence finding. In *17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 359–365, Seattle, Washington, USA, 2001.