



HAL
open science

Sur le choix des abstractions pour le diagnostic hiérarchique

Fabien Perrot

► **To cite this version:**

Fabien Perrot. Sur le choix des abstractions pour le diagnostic hiérarchique. Congrès EDSYS 2007, May 2007, Albi, France. <hal-00170405>

HAL Id: hal-00170405

<https://hal.science/hal-00170405v1>

Submitted on 8 Sep 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

SUR LE CHOIX DES ABSTRACTIONS POUR LE DIAGNOSTIC HIÉRARCHIQUE

Fabien PERROT*

Directrice de thèse : Louise TRAVÉ-MASSUYÈS †

Laboratoire d'accueil :
LAAS-CNRS ‡
7, avenue de Colonel Roche
31077 Toulouse Cedex 4

Établissement d'inscription :
Université Paul Sabatier
118, Route de Narbonne
31062 Toulouse cedex 9

Résumé

Cet article traite du choix des abstractions dans le cadre du diagnostic hiérarchique. Généralement, les abstractions sont choisies à la main et nécessitent une certaine expertise. Nous proposons une solution à ce problème en établissant les critères d'une abstraction qui rend le diagnostic hiérarchique efficace. Pour cela, trois vues complémentaires d'une abstraction sont définies : la vue topologique, la vue dans l'espace d'état et la vue selon les contraintes. Ces trois vues nous permettent d'établir des critères pour trouver l'abstraction la plus efficace afin de résoudre un problème de diagnostic hiérarchique. Cette abstraction correspond au concept de conflit potentiel, bien connu dans la communauté du diagnostic.

Mots-clés

abstraction, diagnostic hiérarchique, conflits, intelligence artificielle.

1 INTRODUCTION

Dans cet article, nous traitons le problème du choix des abstractions pour obtenir des algorithmes de diagnostic hiérarchique efficaces. Nous adoptons la théorie classique en intelligence artificielle du diagnostic basé sur la cohérence (qui est une approche à base de modèles). L'abstraction est, selon de nombreux chercheurs, un des principaux moyens pour améliorer l'efficacité des algorithmes de diagnostic. Le diagnostic hiérarchique consiste en l'utilisation de plusieurs modèles d'un même système pour résoudre le problème de diagnostic. Le diagnostic hiérarchique utilisant l'abstraction structurelle (qui est un cas particulier d'abstraction) a été étudié dans [1]. Cela consiste à regrouper des composants du système pour construire des modèles plus abstraits. Le but de ce travail est de trouver, pour chaque type d'abstraction, parmi tous les modèles abstraits possibles, le plus efficace pour résoudre un problème de diagnostic. Le principal intérêt de cette approche réside dans la conception d'un algorithme qui, à partir d'un modèle détaillé du système, construit un modèle garanti efficace pour le diagnostic sans la nécessité d'une expertise.

*perrot@laas.fr

†louise@laas.fr

‡Ce travail s'effectue dans le cadre d'une collaboration avec le CNES et s'intègre dans le projet AGATA.

Comme certains chercheurs [8], nous voyons le diagnostic comme un problème de satisfaction de contraintes (CSP). En conséquence, les techniques d'abstractions utilisées dans la communauté des contraintes peuvent être appliquées au domaine du diagnostic. L'abstraction de CSPs a été récemment étudiée dans [4] qui ont proposé un cadre formel pour décrire des formes générales d'abstractions, mais à notre connaissance, peu de travaux [7] concernent le choix des abstractions. Cependant l'efficacité des algorithmes de diagnostic hiérarchique dépend de ce choix. Le travail de [7] porte sur les abstractions comportementales. Notre approche porte sur des types différents d'abstractions.

Le reste de cet article est organisé comme suit. Dans la deuxième section, nous présentons brièvement les CSPs et nous rappelons au lecteur comment un problème de diagnostic peut être vu dans ce cadre. Dans la troisième et principale section, nous analysons l'influence du choix des abstractions sur la complexité temporelle des algorithmes hiérarchiques. Finalement, nous concluons et évoquons les travaux futurs.

2 LES CSP ET LES PROBLÈMES DE DIAGNOSTIC

2.1 RAPPELS SUR LES CSP

Un CSP est défini par un ensemble de variables et un ensemble de contraintes sur ces variables. Formellement parlant, une définition d'un CSP est la suivante :

Définition 1 *Un CSP est un couple (X, C) où :*

- $X = \{X_1, \dots, X_n\}$ est un ensemble de variables. Chaque variable X_i a un domaine non vide $D(X_i)$ des valeurs possibles de cette variable.
- $C = \{C_1, \dots, C_m\}$ est un ensemble de contraintes. Chaque contrainte C_j porte sur un sous-ensemble $\text{vars}(C_j)$ de X et spécifie les combinaisons de valeurs autorisées pour ce sous-ensemble de variables.

Un état est une instanciation de toutes les variables de X : $(X_1 = v_1, X_2 = v_2, \dots, X_n = v_n)$. Un état partiel est une instanciation de seulement certaines (mais pas toutes) variables de X : $(X_i = v_k, X_j = v_l, \dots)$.

Une instanciation est dite cohérente si elle ne viole aucune contrainte. Une solution d'un CSP est un état cohérent. Un état est quelquefois appelé instanciation complète et une solution est quelquefois appelée une instanciation cohérente et complète.

La structure d'un CSP est donnée par un hyper-graphe de contraintes où les noeuds de l'hyper-graphe correspondent aux variables du CSP et les hyper-arcs correspondent aux contraintes.

2.2 LES PROBLÈMES DE DIAGNOSTIC DANS LE CADRE DES CSP

Comme certains auteurs [8], nous prétendons que le problème de diagnostic peut être formulé dans le cadre des CSP. Plus précisément, le problème est de trouver une instanciation des variables de modes cohérente avec les contraintes décrivant le comportement du système. Il s'agit d'un CSP où les variables sont séparées en deux sous-ensembles : l'un contenant les variables de décision, et l'autre les variables de non-décision. De plus, une fonction de coût est généralement introduite permettant de comparer les solutions entre elles. Ainsi, la tâche peut être par exemple de trouver les n meilleurs solutions du problème. Ce n'est pas le cas dans cet article. Les modes sont associés aux composants en introduisant des variables de mode que l'on note m_{C_i} pour chaque composant C_i . Par exemple, la formulation dans le cadre des CSP du problème de diagnostic du polybox (un exemple classique dans la communauté du diagnostic DX) est présentée dans la suite.

La topologie du polybox est représentée sur le schéma 1. O_1, O_2, O_3, A_1, A_2 sont les composants du système. O_1, O_2, O_3 sont des portes OU et A_1, A_2 sont des portes ET. Les observables sont instanciés dans cet exemple. X, Y, Z sont des variables non-observables; chacune pouvant prendre ses valeurs dans $\{0, 1\}$. $m_{O_1}, m_{O_2}, m_{O_3}, m_{A_1}, m_{A_2}$ sont les variables de

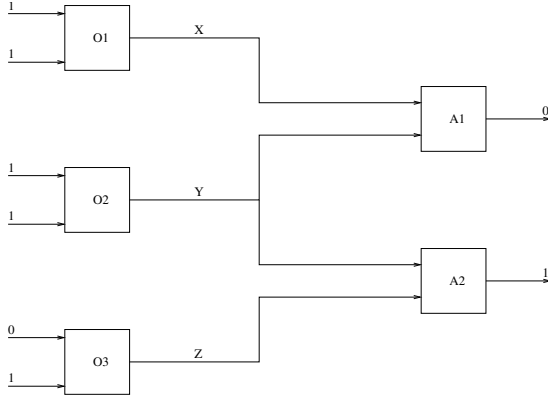


FIG. 1 – Topologie du polybox.

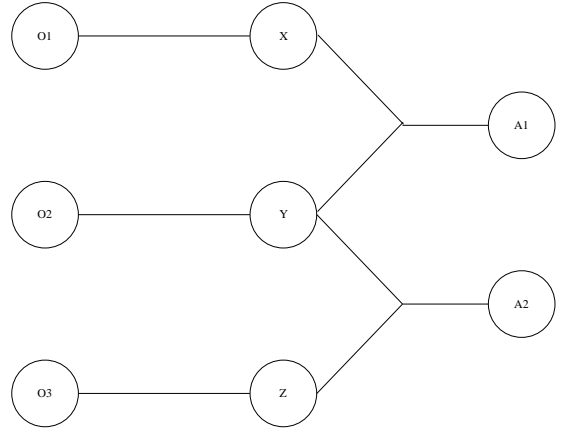


FIG. 2 – Hyper-graphe des contraintes.

modes associées aux composants ; chacune pouvant prendre ses valeurs dans $\{G, B\}$ (G pour Good et B pour Bad). Un ensemble de contraintes lie les variables de modes et les variables non-observables. Chaque contrainte décrit le comportement nominal d'un composant (pour les valeurs des observables données). Pour plus de clarté, nous utilisons un langage de contraintes qui explicite les instanciations (mais le même modèle peut s'écrire en logique propositionnelle par exemple). Les contraintes pour le polybox sont les suivantes :

- $(m_{O_1} = G) \implies (X = (1 \vee 1))$
- $(m_{O_2} = G) \implies (Y = (1 \vee 1))$
- $(m_{O_3} = G) \implies (Z = (1 \vee 0))$
- $(m_{A_1} = G) \implies (0 = (X \wedge Y))$
- $(m_{A_2} = G) \implies (1 = (Y \wedge Z))$

Finalement, le problème de diagnostic du polybox dans le cadre des CSP est :

- variables de décision : $\{m_{O_1}, m_{O_2}, m_{O_3}, m_{A_1}, m_{A_2}\}$, chaque domaine associé est $\{G, B\}$,
- variables de non-décision : $\{X, Y, Z\}$, chaque domaine associé est $\{0, 1\}$,
- les contraintes sont celles décrites ci-dessus.

L'hyper-graphe des contraintes est représenté sur le schéma 2.

3 DIFFÉRENTES VUES DES ABSTRACTIONS

Dans cette section, différentes vues de l'abstraction sont présentées, plus ou moins adéquates selon les propriétés que l'on désire exhiber.

3.1 VUE TOPOLOGIQUE DE L'ABSTRACTION

La vue topologique de l'abstraction est la représentation orientée composant des modèles concrets et abstraits d'un système. Graphiquement, comme montré sur le schéma 3, les composants sont représentés par des noeuds et étiquetés par leur nom. Les liens entre les composants sont représentés par des arcs ou hyper-arcs. Chaque arc est étiqueté par une valeur d'observable ou le nom d'une variable non-observable. Certaines abstractions s'interprètent naturellement dans cette vue et sont faciles à représenter, d'autres peuvent ne pas l'être. Lorsque cette vue est disponible, c'est généralement la plus intuitive.

EXEMPLE

Dans l'exemple du polybox, considérons une abstraction structurelle simple qui consiste à agréger les composants O_1, O_2 et A_1 en un super-composant SC'_1 . La topologie de ce modèle abstrait est représentée sur le schéma 3. Pour définir complètement cette abstraction,

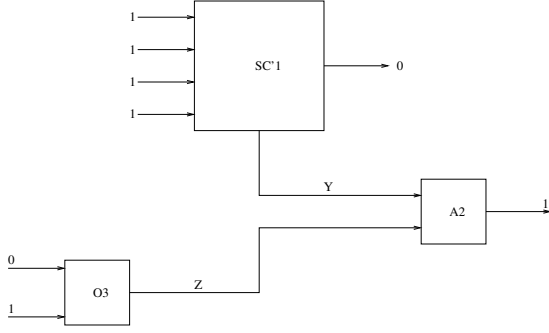


FIG. 3 – Topologie du polybox abstrait.

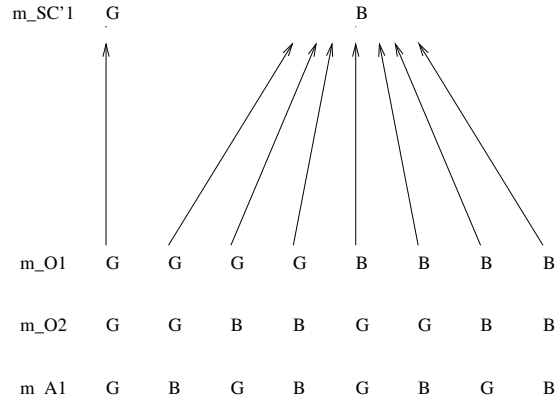


FIG. 4 – Une fonction définie sur $D(m_{O_1}) \times D(m_{O_2}) \times D(m_{A_1}) \rightarrow D(m_{SC'_1})$.

la fonction entre les valeurs possibles $(m_{O_1}, m_{O_2}, m_{A_1})$ et de $m_{SC'_1}$ doit être définie. Par exemple, une fonction naturelle est la suivante :

- si $(m_{O_1} = G, m_{O_2} = G, m_{A_1} = G)$, alors $m_{SC'_1} = G$,
- $m_{SC'_1} = B$ autrement.

La vue topologique ne peut pas capturer toute l'information de l'abstraction car elle ne permet pas de représenter cette fonction.

Cette fonction entre valeurs des variables de modes est représentée sur le schéma 4 et correspond à la projection sur $D(m_{O_1}) \times D(m_{O_2}) \times D(m_{A_1}) \rightarrow D(m_{SC'_1})$ d'une fonction d'interprétation comme définie dans [5]. N'importe quelle fonction parmi les 2^8 possibles fonctions définies sur $D(m_{O_1}) \times D(m_{O_2}) \times D(m_{A_1}) \rightarrow D(m_{SC'_1})$ peut être choisie. Des formes plus générales de relations ont été étudiées dans [4].

3.2 VUE SELON L'ESPACE D'ÉTAT

La vue de l'espace d'état d'un problème de diagnostic est représentée dans l'espace de l'ensemble des instanciations des variables de modes possibles. Une fonction entre valeurs des variables de modes étendue [4] [5] [3] associe des états concrets aux états abstraits. Cette fonction étendue, parce que son domaine est tout l'espace d'état, permet d'évaluer le nombre d'étape nécessaire à un algorithme de diagnostic hiérarchique pour résoudre le problème. Cette vue est donc plus adaptée pour analyser l'influence du choix d'une abstraction sur la complexité temporelle d'un algorithme de diagnostic hiérarchique. Cette vue est plus difficile à représenter sur un schéma car le nombre d'états est exponentiel en fonction du nombre de composants. Sur l'exemple du polybox, les 2^5 états concrets sont abstraits en 2^3 états abstraits.

3.3 VUE SELON LES CONTRAINTES

La vue selon les contraintes d'une abstraction est définie par :

- l'hyper-graphe des contraintes du problème concret et celui du problème abstrait,
- les contraintes des deux problèmes.

Sur l'exemple du polybox, les contraintes 1, 2 et 4 sont abstraites en une seule contrainte : $(m_{SC'_1} = G) \implies ((X = (1 \vee 1)) \wedge (Y = (1 \vee 1)) \wedge (0 = (X \wedge Y)))$. Les contraintes 3 et 5 sont préservées.

4 INFLUENCE DU CHOIX DES ABSTRACTIONS SUR LA COMPLEXITÉ

4.1 ANALYSE DANS LA VUE DE L'ESPACE D'ÉTAT

Parmi toutes les abstractions possibles, deux types généraux se retrouvent dans la littérature [3] [5] [1] [4] :

- les abstractions complètes (CSI pour Concrete Solution Increasing),
- les abstractions correctes (CSD pour Concrete Solution Decreasing).

Les termes complets et corrects sont employés dans le sens suivant : complet désigne le fait que toutes les solutions concrètes sont capturées par les solutions abstraites et correct désigne le fait qu'un état abstrait solution n'a pas d'état concret correspondant non solution.

Définition 2 (Abstraction CSI) *Une abstraction est CSI ssi pour tout état concret solution du problème concret, son état correspondant abstrait est solution du problème abstrait.*

De cette définition, le théorème suivant se déduit trivialement :

Théorème 1 *Si un état abstrait n'est pas solution, alors tous ses états concrets correspondants ne sont pas solutions.*

Par exemple, l'abstraction structurelle du polybox décrite ci-dessus est CSI.

Définition 3 (Abstraction CSD) *Une abstraction est CSD ssi pour tout état concret non-solution du problème concret, son état abstrait correspondant n'est pas solution du problème abstrait.*

De cette définition, le théorème suivant se déduit trivialement :

Théorème 2 *Si un état abstrait est solution, alors ses états concrets correspondant sont solutions.*

Un algorithme de diagnostic hiérarchique simple et général commence par résoudre le problème le plus abstrait, puis se sert des solutions trouvées pour résoudre le problème moins abstrait, et descend ainsi itérativement jusqu'au problème concret. Dans cet article, nous considérons un niveau d'abstraction.

La complexité temporelle de l'algorithme décrit ci-dessus dépend de trois facteurs :

- le type d'abstraction (CSI, CSD, ...),
- le nombre d'états abstraits,
- le nombre d'états concrets.

Une abstraction idéale consiste en deux états abstraits a et b . a est l'image de tous les états concrets solutions, et b est l'image de tous les états concrets non-solutions. Cette abstraction idéale est CSI et CSD.

Théorème 3 *Une abstraction qui est CSI et CSD réduit la complexité temporelle du problème de diagnostic hiérarchique à $O(n')$ où n' est le nombre d'états abstraits. L'abstraction idéale à deux états abstraits.*

L'abstraction idéale ne peut pas être construite dans le cas général pour deux raisons :

- on ne connaît pas à l'avance les solutions et non-solutions,
- les contraintes abstraites ne peuvent pas nécessairement être construites.

Pour la première raison, les solutions sont supposés connues par la suite afin de découvrir quelles abstractions peuvent aisément capturer les solutions et non-solutions. Pour la deuxième raison, la vue selon les contraintes nous permet d'identifier quelles abstractions permettent de construire aisément les contraintes abstraites.

4.2 UTILISER LES CONTRAINTES POUR CHOISIR LES ABSTRACTIONS

L'abstraction idéale peut être approchée en utilisant plusieurs abstractions pour capturer la totalité de l'espace d'état. Pour chaque abstraction, un état abstrait doit représenter un maximum d'états solutions et un autre un maximum d'états non-solutions. L'union des états concrets couverts par chaque état abstrait solution doit couvrir l'ensemble des solutions du problème. De même pour les états non-solutions. Pour représenter un maximum d'état, deux possibilités sont envisageables : agréger les variables et les modes comme décrit dans la vue topologique, ce qui correspond à l'abstraction structurelle ; ou enlever des variables et/ou valeurs. Dans la vue selon les contraintes, agréger des variables correspond à agréger leur contraintes correspondantes ; et inversement. La première option correspond à l'abstraction structurelle, et nous avons vu sur l'exemple du polybox, qu'avec une fonction naturelle, celle-ci est peu efficace dans le cas de deux modes G et B. Pour la seconde option, enlever n variables permet de représenter 2^n états. Enlever le maximum de variables tout en gardant les contraintes testables répond au problème. Le plus petit ensemble de contraintes testables est bien connu en diagnostic puisqu'il correspond aux conflits minimaux potentiels [6] [2]. L'union des états non-solutions capturés par chaque abstraction est bien égale à l'ensemble des états non-solutions du problème car les conflits potentiels minimaux, en les prenant tous, nous garantissent la couverture de l'ensemble de l'espace d'état.

5 CONCLUSION

Dans cet article, la conjecture selon laquelle les abstractions les plus efficaces correspondent à la notion de conflit potentiel est présentée. Pour cela, trois vues originales de l'abstraction ont été définies : la vue topologique, la vue de l'espace d'état et la vue selon les contraintes.

Ce travail a pour but de construire un pont entre les méthodes existantes pour résoudre un problème de diagnostic basées sur les conflits et les méthodes basées sur l'abstraction. Des recherches sont engagées pour consolider ces résultats et étendre cette approche au cadre des systèmes dynamiques et multi-modes.

Références

- [1] L. CHITTARO R. RANON. « Hierarchical model-based diagnosis based on structural abstraction ». *Artif. Intell.*, 155(1-2) :147–182, 2004.
- [2] M.O. CORDIER, P. DAGUE, F. LEVY, J. MONTMAIN, M. STAROSWIECKI L. TRAVÉ-MASSUYÈS. « Conflicts versus analytical redundancy relations. A comparative analysis of the model-based diagnosis approach from the artificial intelligence and automatic control perspectives. ». *IEEE Transactions on Systems, Man and Cybernetics.*, 34(5) :2163–2177, 2004.
- [3] F. GIUNCHIGLIA T. WALSH. « A Theory of Abstraction ». *Artificial Intelligence*, 57(2-3) :323–389, 1992.
- [4] C. LECOUTRE, S. MERCHEZ, F. BOUSSEMART E. GRÉGOIRE. « A CSP Abstraction Framework ». *Revue d'intelligence artificielle*, 20(1) :31–62, 2006.
- [5] P. NAYAK A. LEVY. « A Semantic Theory of Abstractions ». *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.
- [6] B. PULIDO C. ALONSO. « Possible conflicts, ARRs, and conflicts », 2002.
- [7] G. TORTA P. TORASSO. « Qualitative Domain Abstractions for Time-Varying Systems, an Approach based on Reusable Abstraction Fragments ». *Proc 17th Workshop on Principle of Diagnosis DX 06*, June 2006.
- [8] B.C. WILLIAMS R.J. RAGNO. « Conflict-directed A* and Its Role in Model-based Embedded Systems », 2003.