

Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas — une approche fondée sur des techniques d'extraction de connaissances dans des bases de données

Mathieu D'Aquin, Sébastien Brachais, Jean Lieber, Amedeo Napoli

► To cite this version:

Mathieu D'Aquin, Sébastien Brachais, Jean Lieber, Amedeo Napoli. Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas — une approche fondée sur des techniques d'extraction de connaissances dans des bases de données. Rushed Kanawati and Sylvie Salotti and Farida Zehraoui. 12ème Atelier de Raisonement à Partir de Cas - RàPC'04, 2004, Université Paris Nord, Villetaneuse, France, pp.41-52, 2004. <inria-00107765>

HAL Id: inria-00107765

<https://hal.inria.fr/inria-00107765>

Submitted on 19 Oct 2006

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.

Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas — une approche fondée sur des techniques d'extraction de connaissances dans des bases de données *

Mathieu d'Aquin, Sébastien Brachais, Jean Lieber et Amedeo Napoli

LORIA, BP 239, 54 506 Vandœuvre-lès-Nancy
{daquin, brachais, lieber, napoli}@loria.fr

Résumé : Cet article présente une approche pour apprendre automatiquement des connaissances d'adaptation destinées à un système de raisonnement à partir de cas. Cet apprentissage s'appuie sur la base de cas, l'idée étant que, pour deux cas de cette base, leur différence est interprétée comme le résultat d'un processus d'adaptation. La technique d'apprentissage utilisée est l'extraction de motifs fréquents, technique de fouille de données qui suppose un formatage sous la forme d'un ensemble de propriétés booléennes. Le formatage des différences entre cas de la base sous cette forme est discuté en détail, à la fois à un niveau général et au niveau de l'application envisagée, pour le système KASIMIR d'aide à la décision en cancérologie, qui utilise un formalisme proche des logiques de descriptions. L'ensemble du processus d'extraction des connaissances (formatage, fouille et interprétation) est discuté.

Mots-clés : adaptation, apprentissage, extraction de connaissances dans des bases de données, fouille de données, recherche de motifs fréquents

1 Introduction

L'étape d'adaptation du raisonnement à partir de cas (RÀPC) est réputée difficile à implanter. Cela, en particulier, parce qu'elle nécessite des connaissances à acquérir, modéliser et représenter. Cet article propose une approche pour l'acquisition automatique de connaissances d'adaptation (ACA automatique) dans le cadre du RÀPC. Cette approche s'appuie sur des techniques d'extraction de motifs fréquents. La section 2 vise à motiver ce travail en le situant dans le cadre des recherches sur l'adaptation en RÀPC. La section 3 décrit le principe de l'ACA automatique à partir de la base de cas. La section 4 décrit une telle ACA : elle s'appuie sur l'extraction de motifs fréquents, qui est une technique de fouille de données, et sur des principes d'extraction de connaissances dans des bases de données. La section 5 montre comment cela doit s'appliquer au système KASIMIR. La section 6 propose une piste pour évaluer cette approche. Enfin, la section 7 discute de la portée (supposée) de ce travail et de ses extensions possibles.

2 L'adaptation en RÀPC

2.1 Le RÀPC et l'adaptation

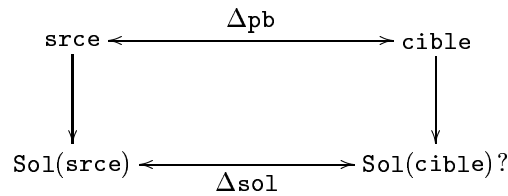
Cette section a pour objectif d'introduire certaines notions relatives au RÀPC. Elle ne prétend pas rendre compte de toutes les variantes de ce mode de raisonnement. Nous considérerons, en particulier, qu'une application du RÀPC présuppose l'existence de deux espaces disjoints :

* Les auteurs tiennent à remercier les deux rapporteurs de cet article pour leurs remarques intéressantes et encourageantes. Cette recherche en cours se fait dans le cadre du projet « Acquisition supervisée et automatique de connaissances d'adaptation » du programme interdisciplinaire TCAN du CNRS, <http://www.dr4.cnrs.fr/tcan/tcan/projets.html>.

Problèmes, l'espace des problèmes et Solutions, l'espace des solutions. Résoudre un problème $pb \in \text{Problèmes}$ consiste à lui associer une solution $Sol(pb) \in \text{Solutions}$. Le RÀPC a pour but de résoudre un problème cible, noté *cible*, en faisant appel à un problème déjà résolu, appelé problème source et noté *srce*, et à sa solution $Sol(srce)$. Les couples $(srce, Sol(srce))$ sont appelés *cas sources* et l'ensemble des cas sources est appelé base de cas. Le raisonnement comprend deux étapes principales : la remémoration et l'adaptation.

La *remémoration* consiste à sélectionner dans la base de cas un cas source $(srce, Sol(srce))$ tel que *srce* et *cible* sont *similaires* (dans un certain sens, qu'il faut définir, mais là n'est pas le propos de cet article).

L'*adaptation* consiste à proposer une solution $Sol(cible)$ au problème *cible*, en modifiant la solution de $Sol(srce)$. Pour ce faire, l'adaptation consiste à effectuer un raisonnement par analogie : « Sachant que $Sol(cible)$ est à $Sol(srce)$ ce que *cible* est à *srce*, connaissant *srce*, $Sol(srce)$ et *cible*, que vaut *cible*? »¹ Le problème d'adaptation peut être « lu » sur le « carré d'analogie » (Chouraqui, 1986) :



Sur ce schéma, Δ_{pb} et Δ_{sol} symbolisent respectivement « ce que *cible* est à *srce* » et « ce que $Sol(cible)$ est à $Sol(srce)$ ». En pratique, Δ_{pb} et Δ_{sol} peuvent prendre de multiples formes. Nous les appellerons *différences* (respectivement entre *srce* et *cible* et entre $Sol(srce)$ et $Sol(cible)$), bien qu'elles peuvent rendre compte à la fois de *similarité* (« ce qui ne varie pas ») et de *dissimilarité* (« ce qui varie »).

L'adaptation peut être schématisée en trois étapes (nous utilisons le symbole \vdash pour « permet d'inférer ») :

- ① $srce, cible \vdash \Delta_{pb}$;
- ② $\Delta_{pb} \vdash_{CA} \Delta_{sol}$ (CA : connaissances d'adaptation disponibles) ;
- ③ $Sol(srce), \Delta_{sol} \vdash Sol(cible)$.

Exemple : Soit la base de cas $\{(srce_1, Sol(srce_1)), (srce_2, Sol(srce_2))\}$, avec $srce_1 = \text{« Recette de la tarte aux pommes ? »}$ et $srce_2 = \text{« Recette de la tarte aux abricots ? »}$, et soit le problème à résoudre $cible = \text{« Recette de la tarte aux poires ? »}$. La remémoration sélectionne le cas $(srce_1, Sol(srce_1))$, si elle juge que, d'un point de vue culinaire, les pommes sont plus proches des poires que les abricots. Une adaptation possible est :

- ① Δ_{pb} : substitution de « pommes » par « poires » (tout restant identique par ailleurs), sachant que les poires sont plus sucrées que les pommes.
- ② Δ_{sol} : substitution de « pommes » par « poires » et diminution de la quantité de sucre (CA doit rendre compte de l'utilité de compenser une augmentation de sucre dans un ingrédient par une diminution de sucre dans un autre).
- ③ $Sol(cible)$ est obtenue en appliquant Δ_{sol} à $Sol(srce)$.

2.2 La place de l'adaptation dans les recherches en RÀPC

Un débat sur l'adaptation a lieu, ou a eu lieu, dans la communauté RÀPC internationale. Les uns considèrent que l'adaptation est nécessairement *ad hoc*, et qu'il est inutile de perdre son temps sur ce problème de recherche. La tentation est forte — et nous n'y résistons pas — de caricaturer cette opinion de la façon suivante : « L'adaptation est à la fois facile à implanter dans un cadre applicatif donné, mais les recherches sur l'adaptation ne mènent jamais à des résultats applicables dans plusieurs domaines d'application. » Les autres considèrent que l'adaptation est centrale (Mille *et al.*, 1996) et que les recherches en RÀPC ne peuvent se concevoir sans prise en compte, sous une forme ou une autre, de l'adaptation. Ce débat s'inscrit dans la distinction entre les approches sans adaptation ou à adaptation faible (p. ex., le *memory-based reasoning* (Stanfill & Waltz, 1986),

1. Ceci pour l'approche transformationnelle de l'adaptation (Carbonell, 1983), il existe également une approche dérivationnelle (Carbonell, 1986).

dont le but est de compenser la pauvreté des adaptations par la richesse de la base de cas) et les approches du *knowledge-intensive* CBR (Aamodt, 1990). C'est à cette deuxième tendance, qui place l'adaptation au centre du RÀPC, que nous adhérons.

Le problème, pour une application donnée, c'est que l'adaptation est réputée difficile à implanter, en particulier si elle est comparée à la remémoration, pour laquelle de nombreuses techniques convaincantes ont été développées. Elle constitue donc un défi pour le chercheur. Pour relever ce défi, trois voies complémentaires pour les recherches sur l'adaptation peuvent être envisagées : les démarches unificatrices, les catalogues et les méthodes d'acquisition de connaissances d'adaptation. Les démarches unificatrices visent à proposer des modèles généraux de l'adaptation sous différents angles (principes, algorithmes, etc.) et peuvent aider un concepteur de système de RÀPC à mieux appréhender l'adaptation dans sa globalité (p. ex. (Fuchs *et al.*, 2000)).

Certains travaux visent à mettre en évidence des catalogues de stratégies d'adaptation qui sont susceptibles de s'appliquer à plusieurs domaines (p. ex. (Riesbeck & Schank, 1989, pp. 41–51) ou (Lieber, 2002)).

Enfin, les démarches d'acquisition de connaissances d'adaptation considèrent l'adaptation dans le cadre d'un domaine d'application donné et visent à mettre en évidence des principes généraux destinés non à l'adaptation elle-même, mais aux moyens de l'explicitier dans le domaine d'application considéré. Nous utiliserons le terme d'ACA (pour *acquisition de connaissances d'adaptation*) pour désigner ces moyens². L'ACA peut être supervisée (citons (Fuchs & Mille, 2000; Lieber *et al.*, 2001) pour des recherches allant vers des méthodologies d'acquisition auprès d'experts) ou automatique (comme nous le verrons ci-dessous). Une ACA semi-automatique, combinant ACA supervisée et ACA automatique, peut également être envisagée.

2.3 L'acquisition automatique de connaissances d'adaptation

Dans (Wilke *et al.*, 1997), un cadre général pour l'ACA automatique est présenté, sous l'aspect des « réservoirs de connaissances » (*knowledge containers*) utiles à une telle acquisition. Nous nous intéressons dans cet article à l'ACA automatique s'appuyant essentiellement sur la base de cas. Plusieurs approches d'une telle ACA ont été proposées dans la littérature (p. ex. (Hanney & Keane, 1996) et (Jarmulak *et al.*, 2001)). Elles sont étudiées comparativement à l'approche décrite ci-dessous et à d'autres approches d'ACA dans (Lieber *et al.*, 2004) (qui peut être considéré comme une annexe du présent article).

Notons que certaines recherches utilisent la base de cas comme source pour une construction automatique de connaissances autres que les connaissances d'adaptation. En particulier, les travaux de (Díaz-Agudo *et al.*, 2003) s'appuient sur des techniques de construction de treillis de Galois pour mettre en évidence des connaissances utiles à l'élaboration du problème cible et à la remémoration. Ces connaissances pourraient avoir un rôle pour l'adaptation, mais les techniques d'apprentissage qui les ont générées n'ont pas été développées dans cette optique.

3 Principe de l'ACA automatique à partir de la base de cas

Le principe envisagé pour l'ACA automatique est illustré par la figure 1. L'idée est d'abord d'extraire des « carrés d'analogie » de la base de cas : étant donné deux cas sources ($srce_1, Sol(srce_1)$) et ($srce_2, Sol(srce_2)$), on modélise Δpb (« différence » entre $srce_1$ et $srce_2$) et Δsol (« différence » entre $Sol(srce_1)$ et $Sol(srce_2)$). Si $\Delta pb \vdash_{CA} \Delta sol$, alors la connaissance d'adaptation disponible CA est suffisante pour résoudre $srce_2$ à partir de ($srce_1, Sol(srce_1)$). Dans le cas contraire, le couple ($\Delta pb, \Delta sol$) est retenu. Avec l'exemple des tartes (cf. section 2.1), Δpb explicite la substitution de « pommes » par « abricots » lors du passage de $srce_1$ à $srce_2$; Δsol explicite les transformations de la recette de la tarte aux pommes vers la recette de la tarte aux abricots (ingrédients, dosages, étapes à ajouter ou à enlever, etc.).

Dans un deuxième temps, à partir de tous les couples ($\Delta pb, \Delta sol$) retenus, sont effectués des regroupements et des généralisations pour arriver à des connaissances d'adaptation, qui sont proposées ensuite à un expert du domaine.

2. Défense de l'acronyme ACA : c'est un palindrome très joli qui résiste assez bien à sa traduction en anglais : AKA.

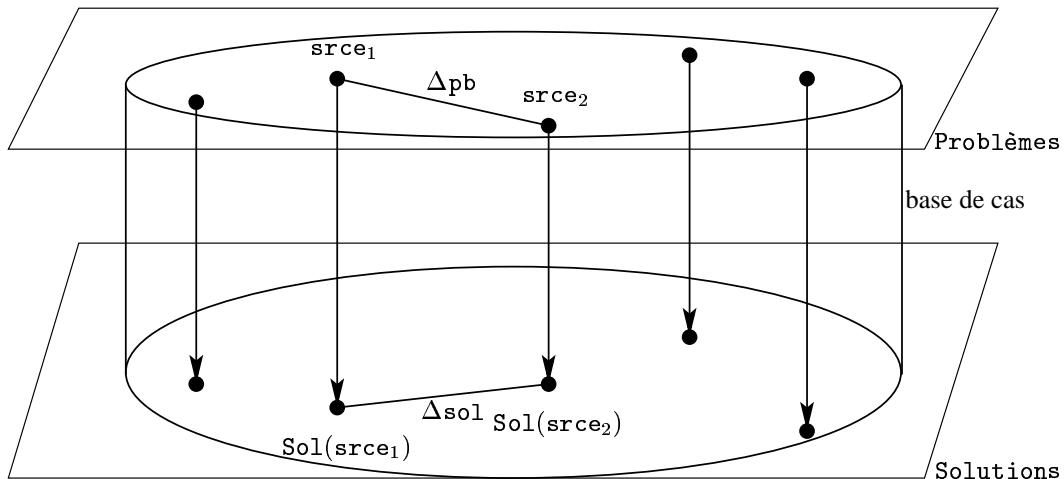


FIG. 1 – La base de cas et la mise en évidence d’un carré d’analogie en son sein (($srce_1, Sol(srce_1)$) et ($srce_2, Sol(srce_2)$) sont deux cas de la base de cas correspondant, dans le carré d’analogie de la section 2.1, respectivement à ($srce, Sol(srce)$) et ($cible, Sol(cible)$)).

Ce principe qui permet de passer de connaissances d’adaptation CA à des connaissances d’adaptation CA’ peut être schématisé par :

- ❶ $srce_1, srce_2 \vdash \Delta pb$;
- ❷ $Sol(srce_1), Sol(srce_2) \vdash \Delta sol$;
- ❸ $CA, (\Delta pb, \Delta sol) \vdash CA'$.

Pour passer de ce principe à sa réalisation, il faut répondre aux questions suivantes :

- Comment modéliser Δpb et Δsol ?
- Faut-il considérer tous les $(\Delta pb, \Delta sol)$ et sinon, lesquels garder ?
- Comment extraire des connaissances d’adaptation à partir de cet ensemble de couples $(\Delta pb, \Delta sol)$?

La section suivante propose une approche pour répondre partiellement à ces questions.

4 ACA automatique fondée sur l’extraction de motifs fréquents

L’approche proposée d’ACA automatique à partir de la base de cas s’appuie sur des principes et des techniques d’extraction de connaissances dans des bases de données (ECBD).

4.1 Rappels sur l’ECBD

L’ECBD a pour objectif de produire des connaissances valides à partir d’un ensemble de données (Bastide *et al.*, 2002). Elle comprend trois étapes principales.

La préparation des données : filtrage et formatage

La préparation des données permet de passer des données brutes et hétérogènes à des données préparées. Le filtrage sert à enlever les données jugées inintéressantes (p. ex., les données bruitées). Le formatage consiste à mettre les données filtrées dans un format uniforme et adapté au(x) processus de fouille. L’application de cette première étape de l’ECBD à l’ACA automatique est décrite à la section 4.2.

La fouille de données

Pour passer des données préparées et formatées à des éléments d'information, un (ou plusieurs) processus de fouille est (ou sont) utilisé(s). Celui que nous allons appliquer à notre problème (à la section 4.3) est l'*extraction de motifs fréquents*. Cette extraction suppose que les données soient représentées par un tableau à deux dimensions de booléens, où les lignes correspondent aux objets de la base $x \in \mathcal{O}$ et les colonnes, aux propriétés p des objets, pris dans un ensemble \mathcal{P} de propriétés. Formellement, les données sont représentées par un triplet $(\mathcal{O}, \mathcal{P}, \mathcal{R})$ où \mathcal{O} est l'ensemble des objets, \mathcal{P} est l'ensemble des propriétés et \mathcal{R} est une relation binaire sur $\mathcal{O} \times \mathcal{P}$: $x \mathcal{R} p$ est vrai signifie que l'objet x a la propriété p . Un motif m est un ensemble de propriétés, i.e., un élément de l'ensemble $2^{\mathcal{P}}$ des parties de \mathcal{P} . Un objet x possède un motif m si pour tout $p \in m$, $x \mathcal{R} p$. Un motif m est fréquent s'il est possédé par beaucoup d'objets de la base, c'est-à-dire si le nombre d'objets qui possèdent m est supérieur à $\sigma_s |\mathcal{O}|$, où $\sigma_s \in [0; 1]$ est un seuil prédéfini. Il existe des algorithmes efficaces pour trouver les motifs fréquents, même pour de très gros volumes de données (voir, en particulier (Bastide *et al.*, 2002)). Par ailleurs, des programmes gratuits implantant ces algorithmes sont disponibles (p. ex. (Weka, 2003)).

L'interprétation

L'objectif de l'interprétation est de construire des unités de connaissances à partir des informations extraites. C'est généralement (toujours ?) le rôle de l'*analyste*, qui est un expert du domaine des données. Cette troisième étape de l'ECBD est envisagée pour notre problème à la section 4.4.

4.2 Filtrage et formatage

Le filtrage consiste à ne garder que les $(\Delta pb, \Delta sol)$ « intéressants ». Peuvent être supprimés d'emblée ceux tels que $\Delta pb \vdash_{CA} \Delta sol$, comme on l'a dit précédemment. Peuvent aussi être éliminés les $(\Delta pb, \Delta sol)$ provenant de deux cas sources jugés trop éloignés l'un de l'autre³.

Le formatage consiste à définir un ensemble de propriétés \mathcal{P} tel que les $x = (\Delta pb, \Delta sol) \in \mathcal{O}$ soient modélisés par des motifs $\Phi(x) \in 2^{\mathcal{P}}$. Un triplet $(\mathcal{O}, \mathcal{P}, \mathcal{R})$ est obtenu de la façon suivante : \mathcal{R} est défini par $x \mathcal{R} p$ ssi $p \in \Phi(x)$. L'approche envisagée est constituée de deux étapes.

Première étape du formatage : associer des propriétés aux problèmes et aux solutions

Nous supposons qu'il existe deux ensembles disjoints de propriétés $\mathcal{P}_{\text{Problèmes}}$ et $\mathcal{P}_{\text{Solutions}}$: le premier contient des propriétés de problèmes et le second, des propriétés de solutions. La première étape du formatage consiste à projeter les problèmes sources et leurs solutions respectivement sur $2^{\mathcal{P}_{\text{Problèmes}}}$ et sur $2^{\mathcal{P}_{\text{Solutions}}}$. Si $(srce, Sol(srce))$ est un cas source, $srce$ sera décrit de façon approchée par $\Phi(srce) \subseteq \mathcal{P}_{\text{Problèmes}}$, l'ensemble des propriétés de $srce$, et $Sol(srce)$ sera décrit de façon approchée par $\Phi(Sol(srce)) \subseteq \mathcal{P}_{\text{Solutions}}$, l'ensemble des propriétés de $Sol(srce)$. Notons que cette première étape du formatage va déterminer les éléments de représentation des cas qui interviendront dans les connaissances d'adaptation acquises. Cette étape va être décrite en détail à la section 5.3, pour l'application KASIMIR.

Deuxième étape du formatage : associer des propriétés aux couples $x = (\Delta pb, \Delta sol)$

Pour associer des propriétés à la différence Δpb entre les problèmes $srce_1$ et $srce_2$ et à la différence Δsol entre les solutions $Sol(srce_1)$ et $Sol(srce_2)$, nous nous sommes inspirés de (Berasaluze *et al.*, 2002), qui propose entre autres une approche pour le formatage des réactions chimiques. Une réaction chimique peut se voir comme la donnée de deux ensembles de molécules : *réactifs* et *produits*. *réactifs* et *produits* sont traduits sous la forme d'ensembles de propriétés chimiques et le formatage de la réaction s'appuie sur ces propriétés et exhibe (1) les propriétés inchangées entre *réactifs* et *produits*, (2) les propriétés créées de *réactifs* à *produits* et (3) les propriétés détruites de *réactifs* à *produits*.

3. Comme le fait remarquer un rapporteur de cet article, la manière de filtrer les couples $(\Delta pb, \Delta sol)$ peut avoir un impact sur les résultats. Ce point devra être examiné soigneusement dans la poursuite de cette recherche, en particulier lors de l'expérimentation.

De façon générale, nous modéliserons la différence entre deux entités a et b par trois ensembles de propriétés : $\Phi^- = \Phi(a) \cap \Phi(b)$, $\Phi^- = \Phi(a) \setminus \Phi(b)$ et $\Phi^+ = \Phi(b) \setminus \Phi(a)$, où $\Phi(u)$ est l'ensemble des propriétés de l'entité u . Ainsi, Δ_{pb} peut être modélisé par les sous-ensembles de $\mathcal{P}_{\text{Problèmes}}$ suivants : $\Phi_{pb}^- = \Phi(\text{srce}_1) \cap \Phi(\text{srce}_2)$, $\Phi_{pb}^- = \Phi(\text{srce}_1) \setminus \Phi(\text{srce}_2)$ et $\Phi_{pb}^+ = \Phi(\text{srce}_2) \setminus \Phi(\text{srce}_1)$. De la même façon, Δ_{sol} peut être modélisé par les sous-ensembles de $\mathcal{P}_{\text{Solutions}}$ suivants : $\Phi_{sol}^- = \Phi(\text{Sol}(\text{srce}_1)) \cap \Phi(\text{Sol}(\text{srce}_2))$, $\Phi_{sol}^- = \Phi(\text{Sol}(\text{srce}_1)) \setminus \Phi(\text{Sol}(\text{srce}_2))$ et $\Phi_{sol}^+ = \Phi(\text{Sol}(\text{srce}_2)) \setminus \Phi(\text{Sol}(\text{srce}_1))$.

Étant donné un couple $x = (\Delta_{pb}, \Delta_{sol})$, six ensembles de propriétés sont ainsi obtenus. On ne peut pas se contenter de faire l'union de ces ensembles de propriétés pour obtenir un motif. En effet, il est important, pour une propriété p , de distinguer son ensemble d'appartenance, parmi les six. Pour distinguer une propriété de problème d'une propriété de solution, il suffit de voir si $p \in \mathcal{P}_{\text{Problèmes}}$ ou si $p \in \mathcal{P}_{\text{Solutions}}$. En revanche, pour $p \in \mathcal{P}_{\text{Problèmes}}$, il n'y a pas de moyen de savoir si p appartient à Φ_{pb}^- , à Φ_{pb}^+ ou à Φ_{pb}^- . L'idée est alors, pour chacune des propriétés $p \in \mathcal{P}_{\text{Problèmes}}$, de créer la propriété p^- si $p \in \Phi_{pb}^-$, p^- si $p \in \Phi_{pb}^-$ ou p^+ si $p \in \Phi_{pb}^+$. On procédera de même pour les $p \in \mathcal{P}_{\text{Solutions}}$. L'ensemble $\Phi(x)$ des propriétés ainsi créées est associé à $x = (\Delta_{pb}, \Delta_{sol})$. Par exemple,

$$\text{si } \begin{cases} \Phi(\text{srce}_1) = \{a, b, c\} & \Phi(\text{Sol}(\text{srce}_1)) = \{A, B\} \\ \Phi(\text{srce}_2) = \{b, c, d\} & \Phi(\text{Sol}(\text{srce}_2)) = \{B, C\} \end{cases} \text{ alors } \Phi(x) = \{a^-, b^-, c^-, d^+, A^-, B^-, C^+\} \quad (1)$$

4.3 Extraction de motifs fréquents

L'extraction de motifs fréquents prend en entrée le triplet $(\mathcal{O}, \mathcal{P}, \mathcal{R})$ et un seuil $\sigma_s \in [0; 1]$, et donne en sortie les motifs $m \in 2^{\mathcal{P}}$ fréquents relativement au seuil σ_s . Supposons par exemple que le motif de (1) soit fréquent. Les propriétés de problèmes Φ_{pb} et les propriétés de solutions Φ_{sol} peuvent être séparées. Ensuite, les règles d'association $\Phi_{pb} \longrightarrow \Phi_{sol}$ peuvent être considérées (Agrawal *et al.*, 1996). Pour l'exemple, cela donnera la règle

$$a^-, b^-, c^-, d^+ \longrightarrow A^-, B^-, C^+ \quad (2)$$

Parmi les règles d'association fréquentes ainsi extraites, sont préférées celles qui ont une *confiance* élevée, sachant que la confiance d'une règle $m_1 \longrightarrow m_2$ est le rapport $\frac{\text{nombre de } x \in \mathcal{O} \text{ possédant } m_1 \text{ et } m_2}{\text{nombre de } x \in \mathcal{O} \text{ possédant } m_1}$ (la confiance est la probabilité conditionnelle que x possède m_2 sachant que x possède m_1 , x étant une variable aléatoire de distribution uniforme sur \mathcal{O}).

Remarque 1

Si $R = \Phi_{pb} \longrightarrow \Phi_{sol}$ est une règle fréquente, une règle R' fréquente ayant même confiance peut être construite, simplement en échangeant les « - » et les « + ». En effet, à chaque $x = (\Delta_{pb}, \Delta_{sol}) \in \mathcal{O}$ (construit à partir de $(\text{srce}_1, \text{Sol}(\text{srce}_1))$ et $(\text{srce}_2, \text{Sol}(\text{srce}_2))$) correspond un $x' = (\Delta_{pb'}, \Delta_{sol}')$ construit à partir des mêmes cas sources, mais considérés dans l'autre sens. Par exemple, la règle correspondant à (2) est :

$$a^+, b^-, c^-, d^- \longrightarrow A^+, B^-, C^- \quad (2')$$

4.4 Interprétation

L'interprétation d'une règle d'association $\Phi_{pb} \longrightarrow \Phi_{sol}$ est généralement effectuée par un expert du domaine, ce qui suppose qu'elle soit présentée de manière intelligible. L'exemple de la règle (2) peut être lu de la façon suivante :

- 1°/ srce et cible partagent les propriétés b et c ,
 - 2°/ la propriété a est dans srce mais n'est pas dans cible ,
 - 3°/ la propriété d n'est pas dans srce mais est dans cible ,
 - 4°/ $\text{Sol}(\text{srce})$ a les propriétés A et B et
 - 5°/ $\text{Sol}(\text{srce})$ n'a pas la propriété C
- alors les propriétés de $\text{Sol}(\text{cible})$ sont $\Phi(\text{Sol}(\text{cible})) = (\Phi(\text{Sol}(\text{srce})) \setminus \{A\}) \cup \{C\}$.

Une telle règle peut donc être vue comme une règle d'adaptation opérant sur les espaces $2^{\mathcal{P}_{\text{Problèmes}}}$ et $2^{\mathcal{P}_{\text{Solutions}}}$. L'interprétation, à l'inverse du formatage, devra traduire ces règles dans les espaces Problèmes et Solutions. Les règles ainsi traduites seront alors examinées par l'analyste. Par ailleurs, il est également nécessaire que l'ensemble des règles présenté à l'analyste soit organisé, par exemple par des regroupements selon différents critères : si beaucoup de règles sont extraites, leur exploration peut être fastidieuse.

Remarque 2

S'il est possible, à partir de toutes les règles d'association $\Phi_{pb} \rightarrow \Phi_{sol}$ ayant une confiance supérieure à un certain seuil et issues de motifs suffisamment fréquents, de construire des règles d'adaptation, les connaissances d'adaptation ainsi acquises sont « symétriques », au sens où, si l'adaptation de $(srce, Sol(srce))$ à *cible* donne $Sol(cible)$, alors l'adaptation de $(cible, Sol(cible))$ à *srce* donnera $Sol(srce)$ (cf. remarque 1). Deux règles d'adaptation « inverses » l'une de l'autre devront être présentées ensemble à l'analyste.

5 Vers une application au système KASIMIR

L'approche proposée ci-dessus doit être validée par une réalisation. Elle va être implantée pour le système KASIMIR. Cette section montre comment l'étape du formatage est envisagée pour cette application.

5.1 Le système KASIMIR

Le système KASIMIR est une plate-forme logicielle développée dans le cadre du projet du même nom (Lieber *et al.*, 2002). Son domaine d'application est l'aide à la décision en cancérologie du sein : un problème cible décrit une patiente et sa maladie (âge, taille de tumeur, etc.), une solution $sol \in \text{Solutions}$ est une proposition de traitement. Les cas sources $(srce, Sol(srce))$ de cette application sont fossilisés (*ossified cases* (Riesbeck & Schank, 1989)); ils peuvent être vus comme des règles : *srce* décrit une classe de patientes et $Sol(srce)$, une proposition de traitement pour les patientes de cette classe. Une application directe de cet ensemble de règles apparaît comme inadéquate dans environ 30% des cas cibles. Pour un tel cas cible, les experts sélectionnent un « cas-règle » $(srce, Sol(srce))$ et l'adaptent. C'est un raisonnement (à partir de cas) que l'on cherche à modéliser informatiquement dans KASIMIR.

5.2 Représentation des cas de KASIMIR

Avant de décrire comment le formatage est envisagé, il est nécessaire de décrire la représentation des problèmes et des solutions qui constituent les cas de KASIMIR.

Le formalisme utilisé par KASIMIR pour représenter les problèmes et les solutions s'apparente aux représentations des connaissances par objets (Napoli *et al.*, 1994) et aux logiques de descriptions (Baader *et al.*, 2003). Un *concept* représente un ensemble d'individus du domaine d'application considéré. La *subsumption* est une relation entre concepts définie par $C_1 \text{ subsume } C_2$ — noté $C_1 \sqsupseteq C_2$ ou $C_2 \sqsubseteq C_1$ — si l'ensemble d'individus représenté par C_1 contient l'ensemble d'individus représenté par C_2 . Si $C_1 \sqsupseteq C_2$ et $C_2 \sqsupseteq C_1$, les deux concepts représentent le même ensemble d'individus ; ils seront dans ce cas considérés comme égaux. \sqsubseteq est une relation d'ordre partiel. Soit \top , le concept qui représente l'ensemble des individus du domaine. \top est donc le maximum pour \sqsubseteq de l'ensemble des concepts. \sqsubseteq permet d'organiser un ensemble de concepts H ayant un maximum ω , en une hiérarchie de racine ω : pour $C, D \in H$, $C \sqsubseteq D$ ssi il existe un chemin dans cette hiérarchie de C à D .

Les briques de base de la représentation dans KASIMIR sont les *concepts primitifs* et les *intervalles numériques*. \top est un concept primitif. Les autres concepts primitifs sont introduits par des liens de subsumption déclarés avec d'autres concepts primitifs. Par exemple, les solutions (traitements) de KASIMIR sont décrits par des concepts primitifs. Une hiérarchie de concepts primitifs est un graphe dont les sommets sont des concepts primitifs comprenant un maximum (p. ex., \top) et les arcs, les liens de subsumption déclarés entre ces concepts primitifs. La figure 2 représente une partie de la hiérarchie des solutions (de racine *traitement*).

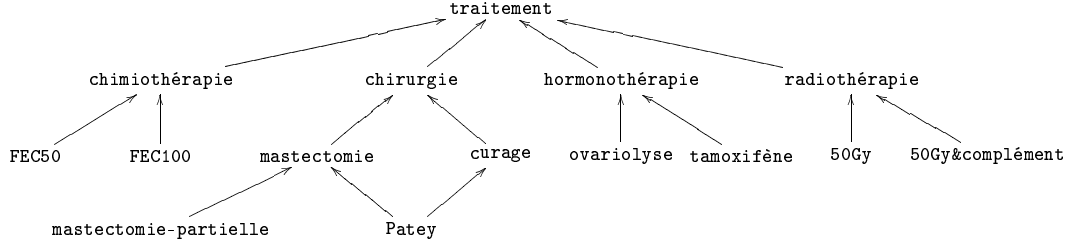


FIG. 2 – Une hiérarchie de concepts primitifs (extrait de la hiérarchie des traitements).

Les intervalles de nombres réels sont partiellement ordonnés par \subseteq . Par exemple, $[2; 4] \subseteq [0; 7]$. Le maximum pour cet ordre est $\mathbb{R} =] - \infty; +\infty[$. Nous considérons dans la suite qu'un intervalle I est un concept ; un individu de l'ensemble représenté par le concept I est un élément de cet intervalle. Par conséquent, $I \subseteq J$ ssi $I \subseteq J$. Dans KASIMIR, en plus des intervalles de réels, sont représentés les intervalles d'entiers et les intervalles angulaires.

Les concepts *définis* de KASIMIR sont de deux types. Le premier type de concept défini est décrit par la donnée d'un *attribut* a et d'un concept C (qui peut être primitif ou défini). Il est noté sous la forme $a : C$. Un tel concept représente l'ensemble des individus liés par une relation binaire a à des individus de l'ensemble représenté par C . Par exemple, $\text{âge} : [7, 77]$ représente l'ensemble des individus dont l'âge est entre 7 et 77 ans.

Le deuxième type de concept défini est la *conjonction* de concepts. Si C et D sont deux concepts, leur conjonction $C \sqcap D$ représente l'intersection des ensembles représentés par C et par D . Notons que dans KASIMIR, les concepts C et D d'une conjonction $C \sqcap D$ doivent être définis. Cependant, cette limitation ne joue pas de rôle particulier dans la suite de cet article.

À titre d'exemple, considérons le concept suivant :

$$C = \text{âge} : [35, 60] \sqcap \text{antécédent} : \text{mastectomie} \sqcap \text{tumeur} : (\text{taille} : [0; 4]) \quad (3)$$

Il représente l'ensemble des individus dont l'âge est entre 35 et 60 ans, ayant eu une mastectomie et dont la tumeur a une taille strictement inférieure à 4 cm.

Enfin, il est important de noter que les définitions circulaires ne sont pas permises dans KASIMIR.

5.3 Le formatage

5.3.1 Spécification

Le formatage, on l'a vu, consiste dans un premier temps, à traduire les problèmes et les solutions représentés dans la base de cas sous la forme d'ensembles de propriétés : au problème srce est associé l'ensemble de propriétés $\Phi(\text{srce}) \in 2^{\mathcal{P}_{\text{Problèmes}}}$ et à la solution $\text{Sol}(\text{srce})$ est associé l'ensemble de propriétés $\Phi(\text{Sol}(\text{srce})) \in 2^{\mathcal{P}_{\text{Solutions}}}$. La deuxième étape du formatage, qui consiste à associer un ensemble de propriétés $\Phi(x)$ à un couple $x = (\Delta_{\text{pb}}, \Delta_{\text{sol}})$, a été décrite en détail à la section 4.3. Nous n'y revenons pas ici.

On dira que p est une propriété de concept si p est une propriété de tous les individus de l'ensemble représenté par ce concept. Soit $\mathcal{P}_{\text{Concepts}}$, un ensemble de propriétés. À $\mathcal{P}_{\text{Concepts}}$ donné, Φ est défini comme étant l'application qui à un concept C associe l'ensemble des $p \in \mathcal{P}_{\text{Concepts}}$ que C possède. Comme les problèmes et les solutions sont représentés par des concepts, définir Φ sur l'ensemble des concepts représentés dans une base de connaissances de KASIMIR (incluant la base de cas) est suffisant. La difficulté consiste donc à définir $\mathcal{P}_{\text{Concepts}}$ (qui contiendra $\mathcal{P}_{\text{Problèmes}}$ et $\mathcal{P}_{\text{Solutions}}$). Pour cela, il faut déterminer ce qu'il est nécessaire de conserver lors de la traduction $\Phi : C \mapsto \Phi(C) \in 2^{\mathcal{P}_{\text{Concepts}}}$.

Or, la relation de *subsumption* entre concepts est essentielle pour KASIMIR ; c'est elle qu'on veut conserver par la traduction. Par ailleurs, la propriété suivante est vérifiée : si $C_1 \sqsupseteq C_2$ alors $\Phi(C_1) \subseteq \Phi(C_2)$. En effet, si $p \in \Phi(C_1)$, p est une propriété commune à tous les éléments de l'ensemble représenté par C_1 , et donc, en particulier, à tous les éléments de l'ensemble représenté par C_2 . La réciproque n'est pas nécessairement vraie, mais nous allons choisir l'ensemble de propriétés

$\mathcal{P}_{\text{concepts}}$ pour que cette réciproque soit vraie pour tous les concepts représentés dans une base de connaissances de KASIMIR. Le formatage Φ doit donc réaliser l'équivalence

$$\text{pour } C_1 \text{ et } C_2, \text{ deux concepts d'une base de connaissances de KASIMIR,} \\ C_1 \supseteq C_2 \Leftrightarrow \Phi(C_1) \subseteq \Phi(C_2) \quad (4)$$

La spécification du formatage est donc la suivante : étant donné un ensemble E de concepts, donner un ensemble de propriétés $\mathcal{P}_{\text{concepts}}$ et une fonction $\Phi : E \rightarrow 2^{\mathcal{P}_{\text{concepts}}}$ réalisant la propriété (4). On supposera que $\top \in E$. Notons qu'une propriété $p \in \mathcal{P}_{\text{concepts}}$ ne sera décrite que par son identificateur (qui sera généré automatiquement).

5.3.2 Algorithme

Le concept \top subsume tous les autres concepts du domaine, par conséquent, $\Phi(\top)$ doit être le minimum pour \subseteq (cf. (4)). Par conséquent, toute propriété $p \in \Phi(\top)$ est partagée par tous les concepts (et donc par tous les individus) de la base de connaissances. Une telle propriété n'est donc pas utile pour le processus d'ACA que nous voulons mettre en place. En accord avec cela, l'algorithme de formatage pose :

$$\Phi(\top) = \emptyset \quad (5)$$

La traduction des concepts primitifs se fait de la façon suivante. À chaque concept primitif C , est associée une propriété de même identificateur que C , qui caractérise sa spécificité. Si on reprend la hiérarchie de la figure 2, au concept primitif *Patey*, est associée la propriété *Patey*. Par ailleurs, pour respecter (4), sont également associées à ce concept les propriétés des concepts qui le subsument. Par conséquent,

$$\text{pour un concept primitif } C \neq \top, \Phi(C) = \{p\} \cup \bigcup_{A: \text{ascendant direct de } C \text{ dans la hiérarchie}} \Phi(A) \quad (6)$$

où p est la propriété définie par le même identificateur que C

Par exemple, s'il est supposé que *traitement* est le descendant direct de \top , on a :

$$\Phi(\text{Patey}) = \{\text{Patey, mastectomie, curage, chirurgie, traitement}\}$$

L'approche pour calculer $\Phi(I)$, où I est un intervalle, est similaire. Prenons par exemple l'intervalle $I = [0; 4[$. L'appartenance d'une variable x à cet intervalle est caractérisée par trois contraintes : $x \in \mathbb{R}$, $x \geq 0$ et $x < 4$. Chacune de ces contraintes va donner naissance à une propriété dont les identificateurs sont *réel*, ≥ 0 et < 4 . Par ailleurs si, dans la base de connaissances, l'intervalle $J = [0; 7[$ a été représenté, comme $I \subseteq J$, on doit avoir $\Phi(I) \supseteq \Phi(J)$. Par conséquent, si $\Phi(J) = \{\text{réel}, \geq 0, < 7\}$, sont associées à I les propriétés qui lui sont spécifiques en plus de celles de J ; s'il n'existe pas d'autres intervalles que J et \mathbb{R} qui contiennent I , on aura $\Phi(I) = \{\text{réel}, \geq 0, < 4, < 7\}$. On suppose que l'intervalle maximal \mathbb{R} est représenté et que :

$$\Phi(\mathbb{R}) = \{\text{réel}\} \quad (7)$$

$$\text{pour un intervalle de réels } I \neq \mathbb{R}, \Phi(I) = F(I) \cup \bigcup_{J \supseteq I, \text{représenté dans la base de connaissances}} \Phi(J) \quad (8)$$

où $F(I)$ est formé des 1 ou 2 propriétés correspondant aux 1 ou 2 bornes finies de I

Pour les intervalles d'entiers ou les intervalles angulaires, la façon de procéder est similaire. En particulier, $\Phi(\mathbb{Z}) = \{\text{entier}\}$.

Le calcul de $\Phi(a : C)$ se fait en construisant, pour chaque propriété $p \in \Phi(C)$, une propriété, dénotée par $a : p$, dont l'identificateur est obtenu par concaténation de l'identificateur de l'attribut a , d'un symbole de séparation (p. ex., « : ») et de l'identificateur de p :

$$\Phi(a : C) = \{a : p \mid p \in \Phi(C)\} \quad (9)$$

À titre d'exemple, supposons que $\Phi([7, 77]) = \{\text{entier}, \geq 7, < 77\}$. On a alors $\Phi(\hat{\text{âge}} : [7, 77]) = \{\hat{\text{âge}} : \text{entier}, \hat{\text{âge}} : \geq 7, \hat{\text{âge}} : < 77\}$.

Pour la conjonction de concepts, on notera que $C \sqcap D \sqsubseteq C$ et que $C \sqcap D \sqsubseteq D$. Par conséquent, $\Phi(C \sqcap D) \supseteq \Phi(C) \cup \Phi(D)$. L'algorithme impose le fait que cette inclusion soit une égalité :

$$\Phi(C \sqcap D) = \Phi(C) \cup \Phi(D) \quad (10)$$

Soit C le concept défini par (3). On suppose que $\Phi([35,60]) = \{\text{entier}, \geq 35, \leq 60\}$, et que $\Phi([0;4]) = \{\text{réel}, \geq 0, < 4, < 7\}$. Dans ce cas, on aura

$$\begin{aligned} \Phi(C) = \{ & \hat{\text{age}}:\text{entier}, \hat{\text{age}}:\geq 35, \hat{\text{age}}:\leq 60, \text{antécédent:mastectomie}, \\ & \text{antécédent:chirurgie}, \text{antécédent:traitement}, \text{tumeur:taille:réel}, \\ & \text{tumeur:taille}:\geq 0, \text{tumeur:taille}:< 4, \text{tumeur:taille}:< 7 \} \end{aligned}$$

Comme les définitions circulaires ne sont pas permises dans KASIMIR, $\mathcal{P}_{\text{Concepts}}$ et Φ peuvent être calculés de la façon suivante. On commence par initialiser $\mathcal{P}_{\text{Concepts}}$ à \emptyset . Puis, pour chaque concept C représenté dans la base de connaissances, on calcule $\Phi(C)$ en utilisant une des équations (5), (6), (7), (8), (9) et (10), selon le type de concept. On ajoute ensuite $\Phi(C)$ à $\mathcal{P}_{\text{Concepts}}$ et on passe à l'itération suivante. On peut montrer, par induction structurelle, que cet algorithme répond à la spécification de la section 5.3.1.

5.4 Un exemple de formatage

Considérons l'exemple suivant (d'après www.oncolor.org):

$$\begin{aligned} \text{srce}_1 &= \text{marges} : [0;0,5[\sqcap N : \text{vrai} \sqcap \text{antécédent} : \text{Patey} \\ \text{Sol}(\text{srce}_1) &= 50\text{Gy} \\ \text{srce}_2 &= \text{marges} : [0;0,5[\sqcap N : \text{vrai} \sqcap \text{antécédent} : \text{mastectomie-partielle} \\ \text{Sol}(\text{srce}_2) &= 50\text{Gy}\&\text{complément} \end{aligned}$$

En supposant que $\Phi([0;0,5[) = \{\text{réel}, \geq 0, < 0.5, < 1\}$, avec $x = (\Delta_{\text{pb}}, \Delta_{\text{sol}})$ construit sur ces deux cas sources, on aura :

$$\begin{aligned} \Phi(x) = \{ & \text{marges:réel}^-, \text{marges}:\geq 0^-, \text{marges}:< 0.5^-, \text{marges}:< 1^-, N:\text{booléen}^-, N:\text{vrai}^-, \\ & \text{antécédent:mastectomie}^-, \text{antécédent:chirurgie}^-, \text{antécédent:traitement}^-, \\ & \text{antécédent:Patey}^-, \text{antécédent:mastectomie-partielle}^+, \\ & 50\text{Gy}^-, 50\text{Gy}\&\text{complément}^+ \} \end{aligned}$$

Ce qui peut s'interpréter par: « si l'antécédent, au lieu d'un Patey (mastectomie totale), est une mastectomie partielle, la radiothérapie subit un complément de dose, dans le contexte $\text{marges} : [0;0,5[\sqcap N : \text{vrai} \sqcap \text{antécédent} : \text{mastectomie}$ » (d'autres contextes donnent les mêmes variations : l'extraction de motifs fréquents devrait permettre de généraliser cette règle).

6 Vers une évaluation de l'approche : reconstruction de la base de cas à partir de fragments

La mise en place d'un test pour cette approche d'ACA automatique est envisagée, pour l'application KASIMIR. Le principe de ce test s'inspire des travaux sur la maintenance de bases de cas qui cherchent à partitionner une base de cas en cas indispensables et en cas optionnels, étant donné les connaissances d'adaptation (voir par exemple (Smyth & Keane, 1995) qui appelle ces cas respectivement *pivotal* et *auxiliary cases* et qui distingue deux catégories intermédiaires). Ce test consistera à examiner dans quelle mesure les connaissances d'adaptation ainsi acquises permettent de reconstituer la base de cas à partir de fragments. Plus précisément, si $(\text{srce}_1, \text{Sol}(\text{srce}_1))$ et $(\text{srce}_2, \text{Sol}(\text{srce}_2))$ sont deux cas sources, on essaie de voir si le second peut être généré à partir du premier et des connaissances d'adaptation acquises CA, selon le schéma suivant :

- ① $CA \vdash (\Delta_{\text{pb}}, \Delta_{\text{sol}})$;
- ② $\text{srce}_1, \Delta_{\text{pb}} \vdash \text{srce}_2$;

③ $\text{Sol}(\text{srce}_1), \Delta_{\text{sol}} \vdash \text{Sol}(\text{srce}_2)$.

Ce qu'on essaiera de tester est quelle proportion de la base de cas peut être reconstituée à partir du reste de la base de cas et des connaissances CA. Ce test soulève des problèmes de spécification et de mise en œuvre que nous n'avons pas encore abordés...

7 Conclusion et perspectives

Cet article propose une approche pour l'ACA automatique à partir de la base de cas. La source de cette adaptation est donnée par des couples de cas sources et par leurs différences. L'algorithme d'apprentissage envisagé est l'extraction de motifs fréquents, technique de fouille de données efficace et qui suppose un formatage sous la forme d'un ensemble de propriétés. Une proposition pour ce formatage est faite à un niveau général, puis spécialisée pour la représentation des connaissances du système KASIMIR, dans un formalisme proche des logiques de descriptions.

La première perspective de recherche va consister à implanter l'approche proposée et à la tester. Les perspectives ci-dessous relèvent donc un peu de la science-fiction, mais comme au moins deux auteurs de cet article apprécient ce genre littéraire, n'hésitons pas⁴ !

Une limitation qu'on peut voir dans l'approche proposée est qu'elle restreint les connaissances d'adaptation acquises au « vocabulaire » de la base de cas (i.e., aux termes atomiques de la représentation des cas sources, à savoir, pour KASIMIR, les concepts primitifs, les bornes des intervalles et les attributs utilisés pour la représentation des cas). En effet, le formatage se fait à partir de l'ensemble fini des problèmes sources et de l'ensemble fini de leurs solutions, et non à partir des espaces Problèmes et Solutions. Pour dépasser cette limitation, on peut envisager une assistance à l'interprétation effectuée par l'analyste qui pousse à abstraire les connaissances d'adaptation acquises. L'étude de la mise en place d'une telle assistance à l'interprétation est une deuxième perspective de recherche.

D'autres approches d'apprentissage que l'extraction de motifs fréquents peuvent également être envisagées. En particulier, les apprentissages qui nécessiteraient un formatage dans un langage moins contraignant devraient nous intéresser. Ils permettraient d'exprimer des différences sous une forme plus expressive : on peut penser aux concepts PPG et DISS décrits dans (Salotti & Ventos, 1999) qui expriment respectivement la similarité et la dissimilarité entre deux concepts exprimés dans une logique de descriptions, voire aux modèles des chemins de similarité et des chemins de modification (Melis *et al.*, 1998), qui expriment respectivement Δ_{pb} et Δ_{sol} . Par exemple, les travaux de (Polailon, 1999) qui visent à étendre les techniques de construction de treillis de Galois pour des données plus complexes que des ensembles de propriétés booléennes, pourraient s'avérer utiles.

Enfin, les auteurs seraient curieux de savoir si l'approche qu'ils proposent pourrait être utile à d'autres applications du RAPC...

Références

- AAMODT A. (1990). Knowledge-Intensive Case-Based Reasoning and Sustained Learning. In L. C. AIELLO, Ed., *Proc. of the 9th European Conference on Artificial Intelligence (ECAI'90)*.
- AGRAWAL R., MANNILA H., SRIKANT R., TOIVONEN H. & VERKAMO A. I. (1996). Fast Discovery of Association Rules. In U. FAYYAD, G. PIATETSKY-SHAPIRO, P. SMYTH & R. UTHURUSAMY, Eds., *Advances in Knowledge Discovery and Data Mining*, p. 307–328, Menlo Park, California: AAAI Press / MIT Press.
- F. BAADER, D. CALVANESE, D. MCGUINNESS, D. NARDI & P. PATEL-SCHNEIDER, Eds. (2003). *The Description Logic Handbook*. Cambridge, UK: Cambridge University Press.
- BASTIDE Y., TAOUIL R., PASQUIER N., STUMME G. & LAKHAL L. (2002). PASCAL : un algorithme d'extraction des motifs fréquents. *Technique et Science Informatiques (TSI)*, **21**(1), 65–95.
- BERASALUCE S., LAURENÇO C. & NAPOLI A. (2002). Extraction de connaissances à partir de bases de données de réactions en chimie organique. In B. BACHIMONT, Ed., *Treizième journées francophones d'ingénierie des connaissances (IC'2002)*, p. 151–162.
- CARBONELL J. G. (1983). Learning by analogy: Formulating and generalizing plans from past experience. In *Machine Learning*, chapter 5, p. 137–161. Tioga Press.

4. Un autre préfère les polars et leur noirceur...

- CARBONELL J. G. (1986). Derivational analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In *Machine Learning*, volume 2, chapter 14, p. 371–392. Springer-Verlag.
- CHOURAQUI E. (1986). Le raisonnement analogique : sa problématique, ses applications. In *Actes des Journées Nationales sur l'Intelligence Artificielle, Aix-les-Bains*, p. 107–117: CEPADUES-Éditions, Toulouse.
- DÍAZ-AGUDO B., GERVÁZ P. & GONZÁLEZ-CALERO P. A. (2003). Adaptation Guided Retrieval Based on Formal Concept Analysis. In *Case-Based Reasoning Research and Development: Proceedings of the Fifth International Conference on Case-Based Reasoning, ICCBR-03*, Lecture Notes in Artificial Intelligence 2689: Springer.
- FUCHS B., LIEBER J., MILLE A. & NAPOLI A. (2000). An Algorithm for Adaptation in Case-Based Reasoning. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000), Berlin, Germany*, p. 45–49.
- FUCHS B. & MILLE A. (2000). Une modélisation au niveau connaissance du raisonnement à partir de cas. In *Actes des journées ingénierie des connaissances (IC-2000)*, p. 3–11, Toulouse.
- HANNEY K. & KEANE M. T. (1996). Learning Adaptation Rules From a Case-Base. In I. SMITH & B. FALTINGS, Eds., *Advances in Case-Based Reasoning – Third European Workshop, EWCBR'96*, LNAI 1168, p. 179–192: Springer Verlag, Berlin.
- JARMULAK J., CRAW S. & ROWE R. (2001). Using Case-Base Data to Learn Adaptation Knowledge for Design. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, p. 1011–1016: Morgan Kaufmann, Inc.
- LIEBER J. (2002). Recopier c'est déjà adapter : six types d'adaptation par copie. In M.-C. JAULENT, C. LE BOZEC & É. ZAPLETAL, Eds., *actes du X^{ème} séminaire français de raisonnement à partir de cas*, p. 11–21.
- LIEBER J., BEY P., BOISSON F., BRESSON B., FALZON P., LESUR A., NAPOLI A., RIOS M. & SAUVAGNAC C. (2001). Acquisition et modélisation de connaissances d'adaptation, une étude pour le traitement du cancer du sein. In *Actes des journées ingénierie des connaissances (IC-2001)*, p. 409–426, Grenoble.
- LIEBER J., D'AQUIN M., BEY P., BRESSON B., CROISSANT O., FALZON P., LESUR A., LÉVÊQUE J., MOLLO V., NAPOLI A., RIOS M. & SAUVAGNAC C. (2002). The Kasimir Project: Knowledge Management in Cancerology. In *Proceedings of the 4th International Workshop on Enterprise Networking and Computing in Health Care Industry (HealthComm 2002)*, p. 125–127.
- LIEBER J., D'AQUIN M., BRACHAIS S. & NAPOLI A. (2004). Une étude comparative de quelques travaux sur l'acquisition des connaissances d'adaptation pour le raisonnement à partir de cas. In SYLVIE SALOTTI, RUSHED KANAWATIET FARIDA ZEHRAOUI, Ed., *à paraître dans les actes du douzième atelier raisonnement à partir de cas, RàPC'04*.
- MELIS E., LIEBER J. & NAPOLI A. (1998). Reformulation in Case-Based Reasoning. In B. SMYTH & P. CUNNINGHAM, Eds., *Fourth European Workshop on Case-Based Reasoning, EWCBR-98*, Lecture Notes in Artificial Intelligence 1488, p. 172–183: Springer.
- MILLE A., FUCHS B. & HERBEAUX O. (1996). A unifying framework for Adaptation in Case-Based Reasoning. In A. VOSS, Ed., *Proceedings of the ECAI'96 Workshop: Adaptation in Case-Based Reasoning*, p. 22–28.
- NAPOLI A., LAURENÇO C. & DUCOURNAU R. (1994). An object-based representation system for organic synthesis planning. *International Journal of Human-Computer Studies*, **41**(1/2), 5–32.
- POLAILLON G. (1999). *Organisation et interprétation par les treillis de Galois de données de type multivalué, intervalle ou histogramme*. Thèse d'université, Université Paris IX-Dauphine.
- RIESBECK C. K. & SCHANK R. C. (1989). *Inside Case-Based Reasoning*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
- SALOTTI S. & VENTOS V. (1999). Approche formelle du raisonnement à partir de cas dans une logique de descriptions. *Revue d'Intelligence Artificielle en Janvier 1999*, **13**, 37–7.
- SMYTH B. & KEANE M. T. (1995). Remembering To Forget. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95), Montréal*, volume 1, p. 377–382.
- STANFILL C. & WALTZ D. L. (1986). Toward memory-based reasoning. *Communications of the ACM*, **29**(12), 1213–1228.
- WEKA (2003). <http://www.cs.waikato.ac.nz/ml/weka/>. dernière consultation : décembre 2003.
- WILKE W., VOLLRATH I., ALTHOFF K.-D. & BERGMANN R. (1997). A Framework for Learning Adaptation Knowledge Based on Knowledge Light Approaches. In *Proceedings of the Fifth German Workshop on Case-Based Reasoning*, p. 235–242.