# Composition et familles génératrices de règles d'adaptation — Application à l'aide à l'interprétation des résultats d'un outil d'extraction de connaissances destiné à un système de raisonnement à partir de cas

Matthieu Tixier*, Fadi Badra*, Jean Lieber*

*LORIA (UMR 7503 CNRS–INPL–INRIA–Nancy 2–UHP)
BP 239, 54 506 Vandœuvre-lès-Nancy, FRANCE
{tixier,badra,lieber}@loria.fr

**Résumé.** L'acquisition de connaissances d'adaptation, notamment dans sa dimension automatique, ouvre de grandes perspectives pour l'étape critique en raisonnement à partir de cas que constitue l'adaptation. Le système CabamakA s'appuie sur des techniques symboliques de fouille de données pour extraire des règles d'adaptation candidates. Cet article présente une approche pour réduire le nombre de telles règles à présenter à l'analyste pour validation. L'idée est de s'appuyer sur une opération de composition de ces règles et de chercher une famille génératrice pour cette composition.

## 1 Introduction

L'adaptation est une étape clé en raisonnement à partir de cas (RàPC), mais nécessite de disposer de connaissances d'adaptation (CA) , ce qui motive les recherches dans le champ de l'acquisition de connaissances d'adaptation (ACA). CabamakA (*Case Base Mining for Adaptation Knowledge Acquisition*) est l'application d'ACA développée dans le cadre du projet Kasimir de gestion des connaissances décisionnelles en cancérologie en vue d'y intégrer un module de RàPC. Inspiré des travaux de Hanney et Keane (1996), et s'appuyant sur des principes et des techniques de l'extraction de connaissances dans des bases de données (ECBD), CabamakA permet d'extraire des règles d'adaptation candidates par fouille de la base de cas. Ces règles doivent ensuite être proposées à un analyste afin de statuer sur leur intégration ou non aux connaissances d'adaptation du module de RàPC. Cette étape d'interprétation et de validation est problématique du fait de l'important volume de règles d'adaptation candidates.

Plusieurs voies de recherches sont ouvertes sur cette problématique d'aide à l'interprétation des résultats de CabamakA (Badra et Lieber (2007)). Ce travail s'inscrit dans la perspective de réduire le nombre de règles à présenter pour validation à l'analyste. Un problème analogue pour les règles d'*association* a donné lieu à de nombreux travaux (voir par exemple Gasmi et al. (2006)). S'inspirant de ce cadre, cet article présente

une notion de composition de règles d'adaptation candidates et de famille génératrice pour cette composition.

Après avoir rappelé les notions de RàPC essentielles à notre propos, nous présenterons les motivations de cette recherche à travers la présentation du système CabamakA et la problématique d'aide à l'interprétation des résultats. Puis nous proposerons dans notre cadre une sémantique pour les règles d'adaptation sur laquelle nous nous appuierons pour définir une loi de composition de règles d'adaptation et également la version faible que nous proposons d'utiliser. Nous présenterons ensuite la notion de famille génératrice sur laquelle nous nous appuyons ainsi que deux algorithmes de construction de telles familles, implantés dans le système GefarX (*GEnerative Family of Adaptation Rules eXtractor*). Les résultats quantitatifs et qualitatifs d'une première étude en vue de valider notre approche seront détaillés afin d'étayer nos conclusions. Nous terminerons en présentant plusieurs perspectives de recherches pour le développement du système GefarX et de notre approche.

## 2 Rappels sur le raisonnement à partir de cas

Raisonner à partir de cas consiste à résoudre un problème à l'aide d'une base de cas, un cas représentant un problème déjà résolu accompagné de sa solution (Riesbeck et Schank (1989)). Un système de RàPC sélectionne un cas dans la base de cas, puis adapte la solution associée. L'adaptation nécessite des connaissances spécifiques au domaine d'application. L'acquisition de connaissances d'adaptation a pour but d'extraire ces connaissances, ce qui peut être réalisé soit directement auprès d'un expert du domaine (d'Aquin et al. (2006)), ou encore par analyse de la base de cas (d'Aquin et al. (2007)).

Un cas est dénoté par un couple (`pb`, `Sol(pb)`) dans lequel `pb` représente un énoncé de problème et `Sol(pb)` une solution de `pb`. L'ensemble des *cas sources* (`srce`, `Sol(srce)`) d'un système de RàPC constitue la *base de cas* `BC`. Lors d'une session particulière de RàPC, le problème à résoudre est appelé *problème cible*, dénoté par `cible`. Un raisonnement à partir de cas associe à `cible` une solution `Sol(cible)`, compte tenu de la base de cas et de bases de connaissances additionnelles.

Le processus de RàPC est principalement composé d'une étape de remémoration et d'une étape d'adaptation. La *remémoration* sélectionne (`srce`, `Sol(srce)`) $\in$ `BC` tel que `srce` est jugé similaire à `cible`. Le but de l'étape d'*adaptation* est de résoudre `cible` en modifiant `Sol(srce)` de façon adéquate. Un *problème d'adaptation* est donné par un triplet (`srce`, `Sol(srce)`, `cible`), et une solution d'un problème d'adaptation est une solution `Sol(cible)` du problème `cible`.

Le modèle d'adaptation adopté est une forme d'*analogie transformationnelle* (Carbonell (1983)) :

1. (`srce`, `cible`) $\mapsto$ $\Delta$`pb`, où $\Delta$`pb` encode les variations entre des problèmes `srce` et `cible`.
2. ($\Delta$`pb`, CA) $\mapsto$ $\Delta$`sol`, où CA est un ensemble de connaissances d'adaptation et $\Delta$`sol` encode les similarités et dissimilarités entre `Sol(srce)` et la solution `Sol(cible)` à construire pour `cible`.

3. (Sol(srce), Δsol) ↦ Sol(cible), Sol(srce) est modifiée en Sol(cible) selon Δsol.

L'étape d'adaptation est dépendante du domaine d'application car elle nécessite des connaissances spécifiques au domaine. Ces connaissances doivent être acquises. C'est l'objet de l'*acquisition de connaissances d'adaptation*.

# 3 Le système CabamakA

## 3.1 Principe

CabamakA (d'Aquin et al. (2007)) est un système d'extraction de connaissances qui permet d'acquérir les connaissances d'adaptation par une analyse systématique des variations entre cas au sein de la base de cas. Le processus d'ECBD mis en œuvre dans CabamakA est constitué d'une étape de préparation des données, d'une étape de fouille puis d'une étape d'interprétation et de validation des résultats par l'analyste.

L'étape de préparation des données applique successivement deux transformations. La *première transformation* Φ transforme les problèmes srce de la base de cas et leurs solutions Sol(srce) en ensembles de propriétés booléennes : $\texttt{srce} \mapsto \Phi(\texttt{srce}) \in 2^{\mathcal{P}}$, $\texttt{Sol(srce)} \mapsto \Phi(\texttt{Sol(srce)}) \in 2^{\mathcal{P}}$. Dans la suite de l'article, nous ne travaillerons que dans le formalisme $2^{\mathcal{P}}$ et nous assimilerons srce et Φ(srce), Sol(srce) et Φ(Sol(srce)).

La *deuxième transformation* génère un ensemble de propriétés booléennes $x$ pour chaque couple de cas sources distincts de la base de cas. Suivant le modèle d'adaptation présenté dans la section 2, cet ensemble de propriétés booléennes encode les variations Δpb et Δsol qui existent lorsqu'on passe du premier cas source au second. Δpb est constitué de l'ensemble des propriétés booléennes présentes dans l'un ou l'autre des problèmes $\texttt{srce}_1$ et $\texttt{srce}_2$, chaque propriété étant marquée :

- d'un « = » si elle est commune à $\texttt{srce}_1$ et $\texttt{srce}_2$,
- d'un « – » si elle est présente dans $\texttt{srce}_1$ mais absente de $\texttt{srce}_2$,
- d'un « + » si elle est présente dans $\texttt{srce}_2$ mais absente de $\texttt{srce}_1$.

Toutes ces propriétés sont reliées à des problèmes et sont marquées par pb. Δsol est calculé de façon similaire et $x = \Delta\texttt{pb} \cup \Delta\texttt{sol}$. Par exemple,

$$\text{si} \quad \begin{cases} \Phi(\texttt{srce}_1) = \{a, b, c, e\} & \Phi(\texttt{Sol}(\texttt{srce}_1)) = \{A, B\} \\ \Phi(\texttt{srce}_2) = \{b, c, d, e\} & \Phi(\texttt{Sol}(\texttt{srce}_2)) = \{B, C, E\} \end{cases}$$
$$\text{alors} \quad x = \left\{ a^{-}_{\texttt{pb}}, b^{=}_{\texttt{pb}}, c^{=}_{\texttt{pb}}, d^{+}_{\texttt{pb}}, e^{=}_{\texttt{pb}}, A^{-}_{\texttt{sol}}, B^{=}_{\texttt{sol}}, C^{+}_{\texttt{sol}}, E^{+}_{\texttt{sol}} \right\} \tag{1}$$

L'étape de fouille consiste à appliquer un algorithme d'extraction de motifs fermés fréquents (MFF) sur les ensembles de propriétés booléennes obtenus. Un tel algorithme s'applique sur un ensemble d'*objets* qui sont décrits par un ensemble d'*attributs*. Un *motif* $m$ est un ensemble d'attributs. Le support d'un motif $m$ est la proportion d'objets $x$ contenant $m$. Un motif est dit *fréquent* si son support est supérieur à un seuil donné. Un motif $m$ est dit *fermé* s'il n'existe pas de motif $m'$ contenant strictement $m$ et de même support.

Dans CabamakA, chaque ensemble de propriétés booléennes $x$ généré pendant la deuxième transformation correspond à la description d'un objet par certains attributs. L'algorithme utilisé pour extraire les motifs fermés fréquents est CHARM (Zaki et Hsiao (2002)), qui est implanté dans la plateforme Coron (Szathmary et Napoli (2005)). Un exemple de MFF généralisant un sous-ensemble d'objets incluant l'objet $x$ donné en (1) est le motif $m_{ex} = \left\{a^{-}_{pb}, b^{=}_{pb}, c^{=}_{pb}, d^{+}_{pb}, A^{-}_{sol}, B^{=}_{sol}, C^{+}_{sol}\right\}$.

### 3.2 L'aide à l'interprétation

La dernière étape de la chaîne de traitement de CabamakA est l'interprétation et la validation des résultats par l'analyste. Dans notre cadre les MFF sont interprétés comme des règles d'adaptation dont la cohérence et la validité doivent être évaluées par l'analyste afin de statuer sur leur intégration ou non aux CA du module de RàPC. Plusieurs recherches sont en cours afin de faciliter l'interprétation des MFF en proposant des outils de navigation ou des métriques de qualité et de pertinence. La recherche présentée ici vise à étudier la structure de l'ensemble des règles d'adaptation extraites afin de réduire le nombre de règles à présenter en validation à l'analyste.

## 4 Motifs extraits et sémantique des règles d'adaptation

De façon générale, une règle d'adaptation (RA) est une application qui permet de résoudre une classe de problèmes d'adaptation (`srce`, `Sol(srce)`, `cible`). On peut donc la voir comme une application *partielle* de l'ensemble des problèmes d'adaptation dans l'ensemble des solutions. Dans cette section, nous nous intéressons aux règles d'adaptations issues de CabamakA : étant donné un MFF $m$, on définit une règle d'adaptation RA($m$), candidate à l'intégration dans les CA du système de RàPC considéré.

Les variations de caractéristiques entre cas sont l'information essentielle pour l'adaptation dans notre cadre. Ces variations sont mises en évidence dans CabamakA à l'aide d'annotations $(-, =, +)$ lors de la deuxième étape du formatage. Soit $m$ un MFF extrait. $m$ est un ensemble de propriétés composé de six sous-ensembles disjoints deux à deux et éventuellement vides, $P^{-}_{\mathtt{pb}}$, $P^{=}_{\mathtt{pb}}$, $P^{+}_{\mathtt{pb}}$, $P^{-}_{\mathtt{sol}}$, $P^{=}_{\mathtt{sol}}$ et $P^{+}_{\mathtt{sol}}$, où $P^{-}_{\mathtt{pb}}$ contient les propriétés de la forme $a^{-}_{\mathtt{pb}}$ (et de même pour les 5 autres ensembles de propriétés). Pour des raisons de lisibilité, nous représenterons un MFF sous la forme d'un tableau :

$$m = P^{-}_{\mathtt{pb}} \cup P^{=}_{\mathtt{pb}} \cup P^{+}_{\mathtt{pb}} \cup P^{-}_{\mathtt{sol}} \cup P^{=}_{\mathtt{sol}} \cup P^{+}_{\mathtt{sol}} = \left|\begin{array}{c|c|c} - & = & + \\ P^{-}_{\mathtt{pb}} & P^{=}_{\mathtt{pb}} & P^{+}_{\mathtt{pb}} \\ P^{-}_{\mathtt{sol}} & P^{=}_{\mathtt{sol}} & P^{+}_{\mathtt{sol}} \end{array}\right|$$

Une sémantique pour les RA est ici présentée en vue de la définition d'une opération de composition des règles d'adaptation. La définition suivante donne une sémantique pour les RA de la forme RA($m$).

**Définition 1** *La règle d'adaptation* RA($m$) *issue du motif* $m = \left|\begin{array}{c|c|c} - & = & + \\ P^{-}_{pb} & P^{=}_{pb} & P^{+}_{pb} \\ P^{-}_{sol} & P^{=}_{sol} & P^{+}_{sol} \end{array}\right|$

*permet d'effectuer le calcul* RA($m$) *:* ($\texttt{srce}$, $\texttt{Sol}(\texttt{srce})$, $\texttt{cible}$) $\mapsto$ $\texttt{Sol}(\texttt{cible})$.

**Si** (1) $P^{-}_{pb}$ *est inclus dans l'ensemble des propriétés propres à* $\texttt{srce}$
(2) $P^{=}_{pb}$ *est inclus dans l'ensemble des propriétés partagées par* $\texttt{srce}$ *et* $\texttt{cible}$
(3) $P^{+}_{pb}$ *est inclus dans l'ensemble des propriétés propres à* $\texttt{cible}$
(4) $P^{-}_{sol}$ *et* $P^{=}_{sol}$ *sont inclus dans l'ensemble des propriétés de* $\texttt{Sol}(\texttt{srce})$
*et* $\texttt{Sol}(\texttt{srce})$ *ne contient aucune propriété de* $P^{+}_{sol}$
**Alors** $\texttt{Sol}(\texttt{cible}) = (\texttt{Sol}(\texttt{srce}) \setminus P^{-}_{sol}) \cup P^{+}_{sol}$

En s'appuyant sur la sémantique des RA($m$), l'analyste est en mesure de valider, rejeter ou modifier ces règles en vue de les intégrer aux CA du système de RàPC. Illustrons cette définition :

**Soit** $m_{ex} = \left\{a^{-}_{pb}, b^{=}_{pb}, c^{=}_{pb}, d^{+}_{pb}, A^{-}_{sol}, B^{=}_{sol}, C^{+}_{sol}\right\} = \left|\begin{array}{c|c|c} - & = & + \\ a & b,\, c & d \\ A & B & C \end{array}\right|$

On considère le problème d'adaptation suivant :
- $\texttt{srce} = \{a, b, c, \alpha_1, \alpha_2\}$
- $\texttt{Sol(srce)} = \{A, B, \Gamma_1, \Gamma_2\}$
- $\texttt{cible} = \{b, c, d, \beta_1, \beta_2\}$

Les $\alpha_i$, $\beta_i$, $\Gamma_i$ sont des propriétés de $\mathcal{P}$ propres au problème d'adaptation considéré et n'apparaissent pas dans $m_{ex}$. RA($m_{ex}$) permet de résoudre ce problème d'adaptation, car (1) $\{a\} \subseteq \texttt{srce}$, (2) $\{b, c\} \subseteq \texttt{srce} \cap \texttt{cible}$, (3) $\{d\} \subseteq \texttt{cible}$ et (4) $\{A\} \cup \{B\} \subseteq \texttt{Sol(srce)}$ et $P^{+}_{\texttt{sol}} \cap \texttt{Sol(srce)} = \emptyset$. L'adaptation par RA($m_{ex}$) donne :
$\texttt{Sol(cible)} = (\texttt{Sol(srce)} \setminus \{A\}) \cup \{C\} = \{B, C, \Gamma_1, \Gamma_2\}$

# 5 Composition de règles d'adaptation

Disposant d'une sémantique pour les règles d'adaptation, on peut s'intéresser à leur composition. L'intuition derrière cette composition est le passage par un problème intermédiaire pb pour résoudre un problème cible en s'appuyant donc, non plus sur une, mais sur deux règles d'adaptation (figure 1).

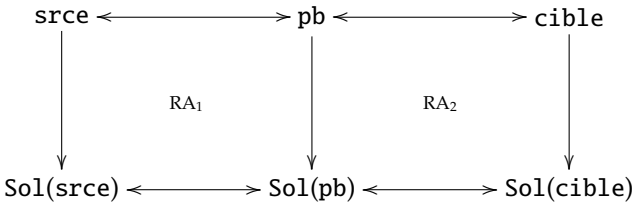


FIG. 1 – *Principe de la composition de* RA.

Le résultat de la composition étant la RA permettant de résoudre directement cible à partir de (srce, Sol(srce)). Ainsi, il n'est plus nécessaire de faire valider par

l'analyste une règle d'adaptation dès lors que l'on peut la reconstruire par composition de règles d'adaptation. La composition de RA, dans le cas général, est définie comme suit :

**Définition 2** *L'opérateur de composition des* RA *est noté « ; ». On a* $\mathrm{RA}_c = \mathrm{RA}_1$ ; $\mathrm{RA}_2$ *si quel que soit* (`srce`, `Sol`(`srce`)) *et* (`cible`, `Sol`(`cible`)) *on a l'équivalence entre les deux affirmations suivantes :*

1. $\mathrm{RA}_c$ *permet d'adapter* (`srce`, `Sol`(`srce`)) *en la solution* `Sol`(`cible`) *de* `cible`*.*
2. *Il existe un problème* `pb` *tel que :*
   (a) $\mathrm{RA}_1$ *permet d'adapter* (`srce`, `Sol`(`srce`)) *en* une *solution* `Sol`(`pb`) *de* `pb`*.*
   (b) $\mathrm{RA}_2$ *permet d'adapter* (`pb`, `Sol`(`pb`)) *en* la *solution* `Sol`(`cible`) *de* `cible`*.*

Dans la suite, nous allons nous intéresser à la composition de règles d'adaptation issues de MFF — $\mathrm{RA}_1 = \mathrm{RA}(m_1)$, $\mathrm{RA}_2 = \mathrm{RA}(m_2)$ — dont le résultat peut aussi s'écrire sous la forme d'un motif — $\mathrm{RA}_c = \mathrm{RA}(m_c)$. La proposition suivante donne une condition *suffisante* pour qu'on ait $\mathrm{RA}(m_1)$ ; $\mathrm{RA}(m_2) = \mathrm{RA}(m_c)$.

**Proposition 1**

$$\textbf{Si} \quad m_1 = \left| \begin{array}{c|c|c} - & = & + \\ P^-_{pb} & P^=_{pb} & P^+_{pb} \\ P^-_{sol} & P^=_{sol} & P^+_{sol} \end{array} \right| \quad \textbf{et} \quad m_2 = \left| \begin{array}{c|c|c} - & = & + \\ Q^-_{pb} & Q^=_{pb} & Q^+_{pb} \\ Q^-_{sol} & Q^=_{sol} & Q^+_{sol} \end{array} \right|$$

Une condition *suffisante* pour que $\mathrm{RA}(m_1)$ ; $\mathrm{RA}(m_2)$ puisse s'écrire sous la forme $\mathrm{RA}(m_c)$ est d'avoir $P^=_{\mathrm{sol}} \cup P^+_{\mathrm{sol}} = Q^-_{\mathrm{sol}} \cup Q^=_{\mathrm{sol}}$. Dans ce cas, on a :

$$m_c = \left| \begin{array}{c|c|c} - & = & + \\ (P^-_{\mathrm{pb}} \cup P^=_{\mathrm{pb}}) \setminus (Q^=_{\mathrm{pb}} \cup Q^+_{\mathrm{pb}}) & (P^-_{\mathrm{pb}} \cup P^=_{\mathrm{pb}}) \cap (Q^=_{\mathrm{pb}} \cup Q^+_{\mathrm{pb}}) & (Q^=_{\mathrm{pb}} \cup Q^+_{\mathrm{pb}}) \setminus (P^-_{\mathrm{pb}} \cup P^=_{\mathrm{pb}}) \\ (P^-_{\mathrm{sol}} \cup P^=_{\mathrm{sol}}) \setminus (Q^=_{\mathrm{sol}} \cup Q^+_{\mathrm{sol}}) & (P^-_{\mathrm{sol}} \cup P^=_{\mathrm{sol}}) \cap (Q^=_{\mathrm{sol}} \cup Q^+_{\mathrm{sol}}) & (Q^=_{\mathrm{sol}} \cup Q^+_{\mathrm{sol}}) \setminus (P^-_{\mathrm{sol}} \cup P^=_{\mathrm{sol}}) \end{array} \right| \quad (2)$$

La preuve de cette proposition est donnée dans Tixier (2007). La proposition 1 introduit une condition suffisante pour la composition de RA sur laquelle il est possible de s'appuyer pour définir un cadre plus restreint nous permettant de travailler sur la composition de RA. L'idée est donc de proposer une composition faible de règles d'adaptation en intégrant cette condition suffisante à ce nouveau cadre.

**Définition 3** *En reprenant les notations de la proposition 1, la composition faible de deux règles d'adaptation* $\mathrm{RA}(m_1)$ *et* $\mathrm{RA}(m_2)$*, notée* $\mathrm{RA}(m_1) \diamond \mathrm{RA}(m_2)$*, est définie si* $P^=_{sol} \cup P^+_{sol} = Q^-_{sol} \cup Q^=_{sol}$ *et est égale à* $\mathrm{RA}(m_c)$ *où* $m_c$ *est défini par l'équation (2). Si cette condition n'est pas vérifiée, on notera* $\mathrm{RA}(m_1) \diamond \mathrm{RA}(m_2) =$ **échec**.

On peut montrer (Tixier (2007)) que $\diamond$ est associative, mais n'est pas commutative et n'admet pas d'élément neutre.

# 6 Famille génératrice et base de règles d'adaptation

La composition faible de RA étant définie, on peut s'attacher à la construction de familles génératrices $G$ telles que leur clôture contient $E_{\mathrm{RA}}$, l'ensemble des RA dérivées des MFF issus de la fouille de données.

**Définition 4** *On appelle clôture pour* $\diamond$ *d'un ensemble* $E_{\mathrm{RA}}$ *de* RA *le plus petit ensemble de* RA *dénoté par* $Clôture_{\diamond}(E_{\mathrm{RA}})$ *tel que :*
- $E_{\mathrm{RA}} \subseteq Clôture_{\diamond}(E_{\mathrm{RA}})$
- **Si** $\mathrm{RA}(m_1)$, $\mathrm{RA}(m_2) \in Clôture_{\diamond}(E_{\mathrm{RA}})$ *et que* $\mathrm{RA}(m_1) \diamond \mathrm{RA}(m_2) \neq$ **échec**, **alors** $\mathrm{RA}(m_1) \diamond \mathrm{RA}(m_2) \in Clôture_{\diamond}(E_{\mathrm{RA}})$

*G est une* famille génératrice *pour* $E_{\mathrm{RA}}$ *si* $Clôture_{\diamond}(G) \supseteq E_{\mathrm{RA}}$. *Par exemple* $E_{\mathrm{RA}}$ *est une famille génératrice finie pour* $E_{\mathrm{RA}}$. *B est une* base *pour l'ensemble* $E_{\mathrm{RA}}$ *si B est une famille génératrice de cardinal minimum pour* $E_{\mathrm{RA}}$. *Notons qu'une base est nécessairement un ensemble fini, puisque* $E_{\mathrm{RA}}$ *est une famille génératrice finie.*

Idéalement, il faudrait étudier comment générer une base de règles d'adaptation pour $\diamond$. Cela constitue une perspective de recherche. Néanmoins, nous avons d'ores et déjà défini et implanté un algorithme de construction de familles génératrices (algorithme 1). Le principe est de construire progressivement la famille génératrice $G$ en ajoutant une à une les règles d'adaptation de l'ensemble initial $E_{\mathrm{RA}}$ jusqu'à ce que la clôture de $G$, qui est construite pas à pas et incomplètement, comprenne $E_{\mathrm{RA}}$.

**Algorithme 1** Construction d'une famille génératrice de règles d'adaptation.

1. $F := E_{\mathrm{RA}}$ (Ensemble des RA non encore couvertes)
2. $G := \emptyset$ (La famille génératrice que l'on cherche à construire)
3. $C := \emptyset$ (La clôture de $G$)
4. **tant que** $C \not\supseteq E_{\mathrm{RA}}$ **faire**
5. Soit $\mathrm{RA} \in F$
6. $F := F \setminus \{\mathrm{RA}\}$
7. $C := C \cup \{\mathrm{RA}\} \cup \{\mathrm{RA} \diamond \mathrm{RA}' \mid \mathrm{RA}' \in C\} \cup \{\mathrm{RA}' \diamond \mathrm{RA} \mid \mathrm{RA}' \in C\}$
8. $G := G \cup \{\mathrm{RA}\}$
9. **fin tant que**
10. **résultat :** $G$

Dans le cadre du système GeFarX, une variante de cet algorithme a été implantée, qui diffère par la façon dont est construit $C$. Dans l'algorithme 1, à chaque itération on ajoute à la clôture l'ensemble des règles résultant des compositions entre la RA courante et l'ensemble des règles de $C$. Ainsi nous sommes amenés à considérer des compositions avec des règles issues elles-mêmes de compositions. Dans ce cadre on ne fixe donc pas de limite au nombre de compositions nécessaires pour trouver les règles participant à l'opération. Dans la deuxième version, on limite le nombre de compositions à une. Les éléments de $C$ sont des éléments de $E_{\mathrm{RA}}$ ou de la composition de deux éléments de $E_{\mathrm{RA}}$. Cela revient à remplacer la ligne 7 de l'algorithme par :

7. $C := C \cup \{\mathrm{RA}\} \cup \{\mathrm{RA} \diamond \mathrm{RA}' \mid \mathrm{RA}' \in G\} \cup \{\mathrm{RA}' \diamond \mathrm{RA} \mid \mathrm{RA}' \in G\}$

Limiter le nombre de compositions permet de réduire la complexité des calculs à effectuer, $|G|$ étant généralement inférieure $|C|$. En contrepartie $C$ est moins exhaustive ce qui oblige à ajouter plus de règles à notre famille génératrice avant de satisfaire la condition d'arrêt. Notons que pour ces algorithmes, la famille génératrice produite est généralement inclue dans $E_{\mathrm{RA}}$.

# 7 Expérimentation

Dans le but de valider notre approche nous avons effectué une première série de tests afin de :

- Mesurer l'efficacité de nos algorithmes en terme de facteur de réduction,
- Comparer leurs temps d'exécution et
- Observer qualitativement les règles d'adaptation pouvant être reconstruites par composition faible d'autres règles.

Les différents algorithmes de construction de familles génératrices sont appliqués sur un ensemble de règles d'adaptation produit par CabamakA. Le domaine applicatif de l'expérimentation est celui de la gestion de connaissances décisionnelles en cancérologie.

## 7.1 Domaine applicatif et jeu de test

Le projet de recherche Kasimir (Lieber et al. (2007)) a pour objet l'aide à la décision et la gestion de connaissances en cancérologie en région Lorraine. L'aide à la décision est l'une des dimensions de ce projet, le principe étant de proposer une recommandation thérapeutique appropriée à partir de la description d'un patient en s'appuyant sur les référentiels de bonnes pratiques utilisés par les professionnels de santé.

L'intégration d'un module de RàPC est une voie de développement intéressante dans l'optique d'être à même de proposer une recommandation lorsqu'un patient ne rentre pas dans le cadre des référentiels. Le référentiel de traitement du cancer du sein peut à cet égard être vu comme une base de cas, dans laquelle un problème `srce` est la description d'une classe de patients et la solution `Sol(srce)` associée à ce problème est une recommandation de traitement. Le référentiel fournit un ensemble de règles $\mathtt{R} \equiv (\mathtt{srce} \longrightarrow \mathtt{Sol(srce)})$ qui couvrent environ 60 à 70% des situations rencontrées. Pour répondre aux situations non prévues par le référentiel, le système de RàPC doit rapprocher le patient hors référentiel d'un cas protocolaire et *adapter* la recommandation thérapeutique standard en s'appuyant sur ses connaissances d'adaptation.

Un premier problème est de trouver un jeu de données de taille limitée permettant de mener à bien nos tests en un temps raisonnable. Le traitement des adénopathies de types inconnu est un ensemble de douze cas extraits du référentiel de traitement du cancer du sein du système Kasimir et est apparu comme un bon candidat pour cette étude.

## 7.2 Protocole expérimental

Pour produire les règles d'adaptation à partir du jeu de données, CabamakA forme tous les $12 \times 11 = 132$ couples de cas sources distincts pris parmi les 12 cas sources sélectionnés, et représente chacun d'eux sous la forme d'un ensemble de propriétés booléennes. Puis l'algorithme CHARM extrait les MFF avec un seuil de support de 1%. Les 784 MFF extraits par CHARM peuvent alors être exploités en tant que règles d'adaptation par nos algorithmes de construction de familles génératrices.

Dix exécutions pour chaque version de l'algorithme ont été effectuées sur cet ensemble de motifs. Pour chacune, on a relevé le facteur de réduction, la taille de $C$ au moment de l'arrêt ainsi que le temps d'exécution.

## 7.3 Résultats quantitatifs

Quel que soit l'algorithme utilisé on parvient à réduire l'ensemble initial de règles d'adaptation d'environ 40 à 60% (tableau 1), ce qui devrait réduire le volume de travail de l'expert de près de la moitié.

| | Réduction (%) | Temps (sec.) | $\|G\|$ | $\|C\|$ |
|---|---|---|---|---|
| Version 1 | $\mu = 59,02$<br>$\sigma = 0,85$ | $\mu = 1880,07$<br>$\sigma = 94,94$ | $\mu = 321,30$<br>$\sigma = 6,63$ | $\mu = 26516,60$<br>$\sigma = 274,51$ |
| Version 2 | $\mu = 39,74$<br>$\sigma = 1,38$ | $\mu = 58,31$<br>$\sigma = 4,28$ | $\mu = 472,40$<br>$\sigma = 10,79$ | $\mu = 16589,60$<br>$\sigma = 275,99$ |

**Tab. 1** – *Comparatif des résultats moyens observés pour la construction de familles génératrices pour chaque algorithme ($\mu$ : moyenne, $\sigma$ : écart-type).*

Le nombre de compositions a une incidence tant sur la performance que sur le temps de calcul. Ainsi on obtient de meilleurs résultats sans limite du nombre de compositions avec près de 60% de réduction, mais cela au prix d'un temps de calcul multiplié par un facteur supérieur à 30. La taille de la clôture est liée à la limite du nombre de compositions. Les variations observées en terme de résultats, de taille de la clôture partielle construite et de temps de calcul sont faibles au sein de chaque banc test. Limiter le nombre de compositions à 1 semble un bon compromis en terme de temps de calcul en l'absence de critères d'ordonnancement permettant d'optimiser la construction des familles génératrices de règles d'adaptation.

## 7.4 Résultats qualitatifs : exemple

$m_1$, $m_2$ et $m_3$ sont trois MFF extraits par CabamakA tels que RA($m_1$) ; RA($m_2$) = RA($m_3$). Cette composition a été extraite des traces de nos algorithmes. Ces règles ont une interprétation dans le domaine médical et expriment des relations entre des nombres de ganglions examinés (nexatt) ou envahis (nenatt) et des recommandations thérapeutiques.

RA($m_1$) met en évidence que pour un patient ayant 4 ganglions envahis, si moins de 10 ganglions ont pu être examinés on ne recommande plus seulement le protocole N+ mais également de faire appel à une RCP pour déterminer le traitement à adopter pour le patient : une RCP (réunion de concertation pluridisciplinaire) réunit des experts médicaux en vue de la résolution de problèmes médicaux particuliers.

$m_1$ =

| − | = | + |
|---|---|---|
| nexatt :supegal10 | indefini<br>nenatt :entiers<br>nenatt :supegal0<br>nenatt :supegal1<br>nenatt :supegal4<br>nexatt :entiers<br>nexatt :supegal0<br>nexatt :supegal1<br>nexatt :supegal4 | nexatt :infegal9 |
| NPSUP3 | INDEFINI<br>NNC<br>NPSUP1 | NPRCP |

RA($m_1$) :

**Si** (1) au moins 10 ganglions du patient `srce` ont été examinés,
(2) au moins 4 ganglions des patients `srce` et `cible` sont envahis et plus de 4 ganglions du patient `cible` ont été examinés,
(3) moins de 9 ganglions du patient `cible` ont été examinés,
(4) la recommandation `Sol(srce)` pour `srce` est le protocole « *N+ : au moins 3 ganglions envahis* » et n'est pas le protocole « *N+ : le nombre de ganglion examiné est insuffisant : demande de RCP* »,
**Alors** `Sol(cible)` = `Sol(srce)` \ {NPSUP3} ∪ {NPRCP}

Le protocole recommandé pour `Sol(cible)` est le protocole « *N+ : au moins 1 ganglion est envahi, le nombre de ganglions examiné est insuffisant : demande de RCP.* »

RA($m_2$) exprime le fait que le nombre de ganglions envahis induit un changement dans le protocole de traitement retenu N+ ou N-. Par ailleurs, si peu de ganglions ont pu être examinés on demande une reprise chirurgicale.

$m_2$ =

| − | = | + |
|---|---|---|
| nenatt :supegal1<br>nenatt :supegal4<br>nexatt :supegal1<br>nexatt :supegal4 | indefini<br>nenatt :entiers<br>nenatt :supegal0<br>nexatt :entiers<br>nexatt :supegal0<br>nexatt :infegal9 | nenatt :infegal3<br>nenatt :infegal9<br>nexatt :infegal3 |
| NPRCP, NPSUP1 | INDEFINI<br>NNC | NMCHIR |

RA($m_2$) :

**Si** (1) plus de 4 ganglions du patient `srce` ont été examinés et plus de 4 ganglions de ce patient sont envahis,
(2) moins de 9 ganglions des patients `srce` et `cible` ont été examinés,
(3) moins de 3 ganglions du patient `cible` ont été examinés et moins de 3 ganglions du patient `cible` sont envahis,
(4) la recommandation `Sol(srce)` pour `srce` est le protocole « *N+ : au moins 1 ganglion envahi, le nombre de ganglions examinés est insuffisant, demande de RCP* » et n'est pas le protocole « *N- : nombre de ganglions examinés insuffisant, reprise chirurgicale* »,
**Alors** `Sol(cible)` = `Sol(srce)` \ {NPSUP1,NPRCP} ∪ {NMCHIR}.

Le protocole recommandé pour `Sol(`*cible*`)` est le protocole « *N- : nombre de ganglions examinés insuffisant, reprise chirurgicale.* »

RA($m_3$) montre également les variations entre un patient sur lequel beaucoup de ganglions ont pu être examinés et où beaucoup sont envahis, et le cas où peu de ganglions ont pu être examinés et peu sont envahis ce qui conduit à un changement de protocole de N+ vers N- et à recommander une reprise chirurgicale.

On a RA($m_1$) ⋄ RA($m_2$) = RA($m_3$). Cette composition est intéressante car elle explicite le mécanisme de l'opération : la condition suffisante de la composition faible est respectée et les sous-ensembles de propriétés exprimant les contraintes sur `srce` et `cible` répartis entre chaque règle sont redistribués pour construire le motif résultant qui ne mentionne plus de problème intermédiaire.

$m_3$ =

| − | = | + |
|---|---|---|
| nenatt :supegal1<br>nenatt :supegal4<br>nexatt :supegal1<br>nexatt :supegal4<br>nexatt :supegal10 | indefini<br>nenatt :entiers<br>nenatt :supegal0<br>nexatt :entiers<br>nexatt :supegal0 | nenatt :infegal3<br>nenatt :infegal9<br>nexatt :infegal3<br>nexatt :infegal9 |
| NPSUP1,NPSUP3 | INDEFINI<br>NNC | NMCHIR |

RA($m_3$) :

**Si** (1) plus de 10 ganglions du patient `srce` ont été examinés et plus de 4 ganglions de ce patient sont envahis,
(3) moins de 3 ganglions du patient `cible` ont été examinés et moins de 3 ganglions de ce patient sont envahis,
(4) la recommandation `Sol(srce)` pour `srce` est le protocole « *N+ : au moins 3 ganglions envahis* » et n'est pas le protocole « *N- : nombre de ganglions examinés insuffisants, reprise chirurgicale* »,

**Alors** `Sol(cible) = Sol(srce)` \ {NPSUP1,NPSUP3} ∪ {NMCHIR}.

Le protocole recommandé pour `Sol(cible)` est le protocole « *N- : nombre de ganglions examinés insuffisant, reprise chirurgicale* ».

# 8 Conclusion et perspectives

Cet article présente une méthode qui vise à réduire le nombre de résultats à présenter à l'analyste lors de l'étape d'interprétation et de validation d'un processus d'ECBD. Cette étude a été menée dans le cadre du développement du système CabamakA d'extraction de connaissances d'adaptation par fouille de la base de cas, qui met en œuvre un algorithme d'extraction de motifs fermés fréquents. Les motifs obtenus sont interprétés comme des règles d'adaptation.

Après avoir défini une loi de composition des règles d'adaptation obtenues à l'issue de l'étape de fouille, nous avons introduit la notion de famille génératrice de règles d'adaptation et proposé des algorithmes permettant de les générer. Des expériences ont montré que la génération de familles de règles d'adaptation permettait de réduire de moitié le nombre de règles à présenter à l'analyste, ce qui devrait réduire d'autant son volume de travail pendant l'étape de validation.

Optimiser le temps de calcul dans le cadre de la construction de clôtures sans limite du nombre de compositions constitue une piste de recherche importante en vue d'une implémentation dans la chaîne de traitements de CabamakA. Par la suite il sera également intéressant de mener une étude théorique des bases pour les règles d'adaptation, notamment en s'inspirant des travaux réalisés dans le cadre des règles d'association comme la base de Duquenne Guigues (Guigues et Duquenne (1986)).

# Références

Badra, F. et J. Lieber (2007). Extraction de connaissances d'adaptation par l'analyse de la base de cas. In *Extraction et gestion des connaissances (EGC'2007), Actes des septièmes journées Extraction et Gestion des Connaissances, Namur, Belgique, 23-26 janvier 2007, 2 Volumes*, Revue des Nouvelles Technologies de l'Information, pp. 751–760.

Carbonell, J. (1983). Learning by analogy : Formulating and generalizing plans from past experience. In R. Michalski, J. Carbonell, et T. Mitchell (Eds.), *Machine Learning :*

*An Artificial Intelligence Approach*, pp. 137–162. Cambridge, MA : Tioga.

d'Aquin, M., F. Badra, S. Lafrogne, J. Lieber, A. Napoli, et L. Szathmary (2007). Case Base Mining for Adaptation Knowledge Acquisition. In M. M. Veloso (Ed.), *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pp. 750–755. Morgan Kaufmann, Inc.

d'Aquin, M., J. Lieber, et A. Napoli (2006). Adaptation Knowledge Acquisition : a Case Study for Case-Based Decision Support in Oncology. *Computational Intelligence (an International Journal) 22*(3/4), 161–176.

Gasmi, G., S. B. Yahia, E. M. Nguifo, et Y. Slimani (2006). IGB : une nouvelle base générique informative des règles d'association. *Revue I3 (Information-Interaction-Intelligence) 6*(1), pp 31–67.

Guigues, J. et V. Duquenne (1986). Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines 95*, 5–18.

Hanney, K. et M. T. Keane (1996). Learning adaptation rules from a case-base. In I. Smith et B. Faltings (Eds.), *Proceedings of the Third European Workshop on Advances in Case-Based Reasoning*, Volume 1168 of *LNAI*, Berlin, pp. 179–192. Springer.

Lieber, J., M. d'Aquin, F. Badra, et A. Napoli (2007). Case-Based Treatment Recommendations for Breast Cancer. *Applied Intelligence (an International Journal)*.

Riesbeck, C. K. et R. C. Schank (1989). *Inside Case-Based Reasoning*. Mahwah, NJ, USA : Lawrence Erlbaum Associates, Inc.

Szathmary, L. et A. Napoli (2005). CORON : A framework for levelwise itemset mining algorithms.

Tixier, M. (2007). Composition et familles génératrices de règles d'adaptation. Rapport de M2R, master SCA, spécialité TAL, université Nancy 2.

Zaki, M. J. et C.-J. Hsiao (2002). CHARM : An efficient algorithm for closed itemset mining. In R. L. Grossman, J. Han, V. Kumar, H. Mannila, et R. Motwani (Eds.), *SDM*. SIAM.

## Summary

Adaptation is a crucial but difficult step to achieve in Case Based Reasoning systems due to the need for domain specific knowledge that is hard to acquire. One way to achieve it is to make use of knowledge discovery techniques to automatically acquire domain knowledge from the case base. This is the purpose of the system CabamaKA, which extracts a set of adaptation rules from the case base by an analysis of the variations that exist between cases. The proposed approach aims at reducing the set of rules that are presented to the analyst for validation. It is achieved by defining a composition operation on these rules and computing a generative family of adaptation rules for this operation.