



**HAL**  
open science

## Une première formalisation de la phase d'élaboration du raisonnement à partir de cas

Jean Lieber, Béatrice Fuchs, Alain Mille, Amedeo Napoli

► **To cite this version:**

Jean Lieber, Béatrice Fuchs, Alain Mille, Amedeo Napoli. Une première formalisation de la phase d'élaboration du raisonnement à partir de cas. 14ème atelier francophone de raisonnement à partir de cas, Mar 2006, Besançon, France. <inria-00110932>

**HAL Id: inria-00110932**

**<https://inria.hal.science/inria-00110932v1>**

Submitted on 2 Nov 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.



HAL Authorization

# Une première formalisation de la phase d'élaboration du raisonnement à partir de cas

Béatrice Fuchs<sup>1</sup>, Jean Lieber<sup>2</sup>, Alain Mille<sup>1</sup>, Amedeo Napoli<sup>2</sup>

<sup>1</sup> LIRIS CNRS, UMR 5202, Université Lyon 1, INSA Lyon, Université Lyon 2, ECL  
Alain.Mille@univ-lyon1.fr

<sup>2</sup> LORIA UMR 7503, CNRS, INRIA, Universités de Nancy  
Jean.Lieber@loria.fr

---

**RÉSUMÉ :** La phase d'élaboration du raisonnement à partir de cas a pour objectif de préparer la remémoration en enrichissant la description du problème tel qu'il est saisi par l'utilisateur, afin d'obtenir le problème cible. Cet article tente une première formalisation de l'élaboration en s'appuyant sur un exemple et sous l'hypothèse, assez générale, selon laquelle les problèmes et solutions peuvent être considérés comme des ensembles (ou conjonctions) de descripteurs. Cette formalisation est une définition qui caractérise une fonction d'élaboration. Plusieurs fonctions d'élaboration sont possibles, de celle qui consiste à ne rien faire (le problème cible est le problème saisi par l'utilisateur) à celle qui consiste à ajouter tous les descripteurs déductibles du problème saisi et de la théorie du domaine. Un cas intermédiaire intéressant consiste à ne rajouter au problème saisi par l'utilisateur que les descripteurs susceptibles de jouer un rôle dans le raisonnement, i.e., ceux qui sont liés à des dépendances entre problèmes et solutions.

**MOTS-CLÉS :** élaboration, formalisation, conjonction de descripteurs, dépendances

---

## 1. Introduction

Le raisonnement à partir de cas (RÀPC (Riesbeck *et al.*, 1989)) peut être vu comme un mode de résolution de problèmes s'appuyant sur une base de cas, un cas étant la représentation informatique d'un épisode de résolution de problèmes. Le problème à résoudre est appelé problème cible, noté *cible*, un cas de la base de cas est un cas source, noté sous la forme  $(srce, Sol(srce))$  où *srce* est un *problème source* et où  $Sol(srce)$  est la solution associée à *srce* (généralement, d'autres informations sont associées à un tel cas source, notamment des informations sur la résolution de problème  $srce \mapsto Sol(srce)$ ). Ce mode de raisonnement se décompose souvent en trois étapes principales : la remémoration, l'adaptation et la mémorisation. La remémoration consiste à chercher un cas source  $(srce, Sol(srce))$  tel que *srce* est jugé similaire à *cible*. L'adaptation consiste à construire ou trouver une solution  $Sol(cible)$  de *cible* en s'appuyant sur le cas remémoré. La mémorisation consiste à stocker le nouveau cas  $(cible, Sol(cible))$  une fois celui-ci validé, si ce stockage est jugé opportun.

Avant l'étape de remémoration (ou dans une première phase de la remémoration, si on préfère), a lieu l'*élaboration*. Cette étape consiste, à partir de l'*entrée* du système de RÀPC, à construire le problème cible :

$$\text{Élaboration : pré-cible} \longmapsto \text{cible}$$

où *pré-cible* dénote l'entrée du système de RÀPC, c'est-à-dire, si on veut, le problème cible

avant mise en forme. Cette étape d'élaboration a été présentée dans (Herbeaux, 2000). Lors du 13<sup>ème</sup> atelier francophone sur le RÀPC (RÀPC, 2005), a eu lieu une table ronde sur l'élaboration, menée par A. Mille. Cet article fait suite à cette table ronde et à une discussion entre les auteurs de cet article qui a permis de converger vers une idée commune sur la formalisation de ce qu'est la notion d'élaboration. Le résultat est sous vos yeux, c'est un article écrit en français, avec une police de caractère choisie pour en faciliter la lecture.

Dans un premier temps, quelques rappels généraux sur le RÀPC sont faits et des notations sont introduites (section 2). Puis, l'élaboration est présentée informellement dans la section 3, qui justifie son importance au sein du cycle de RÀPC. La section 4 débute par l'introduction d'un exemple d'élaboration. Cet exemple permet d'introduire une première définition formelle de l'élaboration qui donne des caractéristiques sur ce que doit être une fonction d'élaboration. Cette définition est alors utilisée pour décrire diverses fonctions d'élaboration. La section 5 discute des liens entre l'élaboration et le principe d'indépendance à la syntaxe. Enfin, la section 6 présente une conclusion en soulignant principalement l'intérêt d'étudier l'élaboration ainsi que des perspectives de ce travail.

## 2. Rappels sur le RÀPC

On se place dans un domaine d'application particulier, pour lequel les notions de problème et de solution sont bien définies. Soit *Problèmes* et *Solutions* l'ensemble des représentations des problèmes et des solutions de ce domaine : si  $pb \in \text{Problèmes}$  (resp.,  $sol \in \text{Solutions}$ ) alors  $pb$  (resp.,  $sol$ ) est une expression d'un formalisme de représentation des connaissances qui représente un problème (resp., une solution) de ce domaine. Par ailleurs, on suppose qu'il existe une relation binaire sur  $\text{Problèmes} \times \text{Solutions}$  dont la signification est « a pour solution ». Dans certaines applications du RÀPC, cette fonction n'est qu'imparfaitement connue. En revanche, on dispose d'une base de cas (appelés cas sources), c'est-à-dire d'un ensemble de couples  $(srce, \text{Sol}(srce))$  où  $srce \in \text{Problèmes}$ ,  $\text{Sol}(srce) \in \text{Solutions}$  et  $srce$  a pour solution  $\text{Sol}(srce)$ .

On appelle  $\text{Th}$ , la *théorie du domaine* : c'est une base de connaissances relative au domaine d'application.  $\text{Th}$  concerne en particulier les éléments de descriptions des problèmes et des solutions. Étant donné deux expressions  $e_1$  et  $e_2$ , on notera  $e_1 \stackrel{\text{Th}}{\equiv} e_2$  si ces deux expressions sont sémantiquement équivalentes (pour tout modèle  $\mathcal{I}$  de la théorie  $\text{Th}$ ,  $e_1^{\mathcal{I}} = e_2^{\mathcal{I}}$ ).  $\text{Th}$  est supposée contenir toute la connaissance du système de RÀPC qui est en-dehors de la base de cas : elle contient donc la connaissance de remémoration (ou similarité) et la connaissance d'adaptation.

Une façon fréquente de représenter un problème en RÀPC est sous la forme d'un ensemble de *descripteurs* :

$$pb = \{d_1, d_2, \dots, d_n\} \quad (1)$$

Un descripteur est une expression du formalisme de représentation (elle peut être atomique, sous la forme d'un couple attribut-valeur, sous la forme d'une contrainte, etc.). Cette description s'interprète comme une conjonction :  $pb$  est décrit par  $d_1, d_2, \dots$  et  $d_n$  ; on notera  $pb$  indifféremment sous la forme (1) ou sous la forme

$$pb = d_1 \wedge d_2 \wedge \dots \wedge d_n \quad (2)$$

Ce formalisme général va être utilisé dans cet article. On note *Descripteurs*, l'ensemble (potentiellement infini) des descripteurs. *Problèmes* est alors l'ensemble des représentations sous

la forme d'ensembles de descripteurs :  $\text{Problèmes} = 2^{\text{Descripteurs}}$ . Notons que rien n'empêche théoriquement un problème d'être décrit par une infinité de descripteurs, mais que cela pose un problème pratique ! Nous y reviendrons au cours de l'article.

### 3. Présentation générale de l'élaboration

La remémoration de cas sources se fait en général à partir d'une mesure de similarité. Cette mesure de similarité exprime l'importance et l'influence des descripteurs problèmes pour sélectionner un cas dont la solution sera adaptable en vue de la résolution d'un problème cible.

Prenons l'exemple d'un problème d'estimation du prix d'un véhicule d'occasion : il apparaît rapidement que la connaissance du kilométrage d'un véhicule est plus importante que la connaissance de la couleur des sièges pour adapter le prix d'un véhicule à un autre. Les connaissances permettant de pondérer les descripteurs de problème sont de même nature que celles qui permettent d'adapter les descripteurs de solution. En effet, dans les deux cas, il s'agit de connaître la dépendance et l'influence sur la solution des descripteurs du problème. Si, dans certains domaines, les facteurs influents peuvent être les mêmes quels que soient les cas de la base, il arrive fréquemment que les facteurs influents dépendent des cas. Par exemple, dans le cas d'un véhicule de collection, il est clair que la couleur d'un siège (rare par exemple) va influencer beaucoup plus que le nombre de kilomètres sur le prix !

C'est le rôle de l'élaboration de construire les descripteurs influents pour que la recherche de cas proches et par suite l'adaptation aient le plus de chances d'être efficaces.

#### 3.1. Quelques travaux impliquant l'élaboration dans la littérature

La littérature propose différentes voies pour mener cette élaboration sans toutefois la nommer ainsi, ni en faire une étape en soi du raisonnement à partir de cas.

(Cain *et al.*, 1991) proposent par exemple de représenter un cas comme un ensemble de faits décrivant un problème, un ensemble de faits décrivant une solution et un ensemble d'explications reliant problème et solution. Un fait  $f_i$  est considéré comme descripteur problème s'il est intervenu comme prémisse d'une explication d'un cas passé. (Bento *et al.*, 1993) ont amélioré cette technique pour tenir compte de la « qualité » des explications. Les faits candidats pour être descripteurs problèmes deviennent alors plus ou moins pertinents selon qu'ils sont prémisses d'explications complètes, partielles ou incomplètes.

Une autre réponse est proposée par A. Goel avec le système KRITIK (Goel *et al.*, 1997) ou encore R. J. Aarts dans le domaine de la planification de process biologiques (Aarts *et al.*, 1996; Rousu *et al.*, 1996). Dans un premier temps ces systèmes extraient de la base de cas un ensemble de cas en se fondant sur une similarité de description de problème, puis dans un deuxième temps ils évaluent le coût de l'adaptation sur les cas les plus proches du problème spécifié. La sélection par mesure de similarité de cas « proches » exclut un certain nombre de cas qui auraient peut-être (en y regardant de près) été intéressants. Travailler sur des cas élaborés immédiatement en prenant en compte cette connaissance serait moins risqué et tout aussi efficace.

Une autre piste a été explorée dans le système Déjà Vu (Smyth *et al.*, 1996) : celle appelée *recherche guidée par l'adaptation*. Le principe est de trouver un critère capable de prédire l'adaptabilité d'un cas sans avoir à l'adapter complètement. Pour y parvenir, Déjà Vu s'appuie sur des *spécialistes* et des *stratégies*. Les *spécialistes* sont composés de règles *condition-action*.

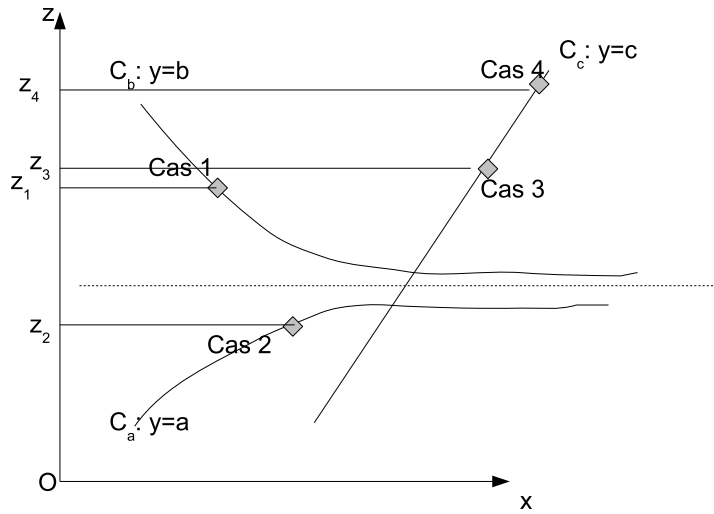


FIG. 1 – Illustration de trois classes d’adaptations disjointes nécessaires à repérer au moment de l’élaboration.

La partie condition précise le type de différence qu’il est capable de réduire, tandis que la partie action est constituée d’opérateurs permettant de transformer les graphes solution (substitution, insertion, suppression d’éléments du graphe). Les *spécialistes* d’adaptation sont associés à un *coût* fonction du type d’opération menée. Comme les spécialistes agissent localement, des problèmes de conflit peuvent survenir entre différentes actions qu’ils mènent séparément. C’est pour régler cette difficulté que Déjà Vu exploite les *stratégies* pour sélectionner les cas les plus adaptables, c’est-à-dire présentant un faible coût d’adaptation et une faible possibilité de conflit. Les connaissances du domaine contenues dans ces stratégies sont typiques de ce que nous appelons « connaissances d’élaboration ».

### 3.2. Illustration de l’importance de l’élaboration

Quand un critère d’appréciation de l’utilité d’un cas pour son adaptation à un cas cible a été défini, il faut rechercher un principe qui permettra de retrouver les cas les plus utiles. La figure 1 présente une situation simplifiée où un descripteur  $z$  dépend d’un descripteur  $x$ . La manière dont  $z$  varie en fonction de  $x$  (de façon croissante, décroissante, linéaire, etc.) dépend de la valeur de  $y$ .  $x$  est de type numérique et  $y$  de type symbolique avec les valeurs  $\{a,b,c\}$ . Les connaissances d’adaptation sont représentées sur la figure 1.

Bien que les solutions  $z_1$  et  $z_3$  soient très proches, elles ne sont pas adaptables de la même façon. Il est nécessaire d’élaborer le problème cible avec un descripteur inféré de telle sorte qu’il garantisse la différenciation des méthodes d’adaptation correspondant aux différentes valeurs de  $y$ . La méthode générale d’adaptation consiste donc à générer des descripteurs *indices d’adaptabilité* qui compléteront la description initiale du problème.

### 3.3. Profil de l’élaboration

Nous considérerons dans cet article que l’élaboration est une étape qui à un problème représenté par pré-cible, l’entrée du système de RÀPC, associe le problème cible *élaboré*, noté cible :

$$\text{Élaboration : pré-cible} \in \text{Problèmes} \mapsto \text{cible} \in \text{Problèmes} \quad (3)$$

On définit informellement l'élaboration comme une étape du RÀPC qui a pour objectif de *préparer* la remémoration en *enrichissant* la description du problème posé au système.

Mais sur quelle base peut-on enrichir la description de pré-cible ? C'est ce à quoi nous tâcherons de répondre à la section suivante, après l'intermède ci-dessous.

*Évacuons le problème annexe de l'élaboration du cas source.* Il existe éventuellement une autre étape d'élaboration : celle du cas source entré par le concepteur de la base de cas initiale. En fait, l'idée est que la saisie d'un tel cas source (y compris le tout premier des cas sources) peut suivre le processus du RÀPC : on décrit un problème pb, avec une remémoration qui pourra ne donner aucun cas jugé similaire et une adaptation qui consistera simplement à demander au concepteur de décrire la solution  $Sol(pb)$  de ce problème. Le couple  $(pb, Sol(pb))$  sera alors stocké dans la base de cas, comme cas source. L'élaboration de ce cas source correspondra alors à la même étape d'élaboration que l'élaboration du problème pré-cible, à savoir au processus  $Élaboration(\cdot)$  qui aura permis de décrire pb.

## 4. Exemple et première formalisation

### 4.1. Présentation et étude de l'exemple

Considérons le domaine dans lequel un problème est la description d'une voiture et la solution est un prix « raisonnable » pour cette voiture. Supposons que l'utilisateur d'un système de RÀPC sur ce domaine décrive une 206 coupée de 10 ans ayant roulé 198 000 km et ayant comme seul défaut une corrosion superficielle. Cette description se traduit dans un formalisme d'ensemble de descripteurs où un descripteur est un couple attribut a - valeur v, noté  $(a = v)$ , de la façon suivante :

$$\begin{aligned} \text{pré-cible} &= (\text{âge} = 10) \\ &\wedge (\text{kilométrage} = 198\,000) \\ &\wedge (\text{listeDéfauts} = (\text{"corrosion superficielle"})) \\ &\wedge (\text{modèle} = \text{"206"}) \\ &\wedge (\text{type} = \text{"coupé"}) \end{aligned}$$

Si on dispose des connaissances selon lesquelles (1) si une voiture a une liste (exhaustive) de défauts contenant moins de 3 éléments, alors elle a un bon état général et (2) 206 est un modèle de la marque Peugeot, on peut compléter la description du problème de la façon suivante :

$$\begin{aligned} \text{cible} &= \text{pré-cible} \\ &\wedge (\text{étatGénéral} = \text{"bon"}) \\ &\wedge (\text{marque} = \text{"Peugeot"}) \end{aligned}$$

Les connaissances sur lesquelles on s'est appuyées font partie de la théorie du domaine :

$$\begin{aligned} Th = \{ &(\text{longueur}(\text{listeDéfauts}) < 3) \Rightarrow (\text{étatGénéral} = \text{"bon"}), \\ &(\text{modèle} = \text{"206"}) \Rightarrow (\text{marque} = \text{"Peugeot"}) \dots \} \end{aligned}$$

Cet exemple peut être examiné selon les points de vue sémantique et syntaxique.

Selon le point de vue sémantique, étant donné la théorie du domaine  $Th$ , rien ne permet de distinguer pré-cible et cible : pré-cible  $\stackrel{Th}{\equiv}$  cible. En généralisant pré-cible à n'importe quel  $pb \in \text{Problèmes}$ , on obtient :

$$\text{Élaboration}(pb) \stackrel{Th}{\equiv} pb \quad (4)$$

Autrement dit, l'élaboration est une opération qui ne modifie pas le sens du problème sur lequel elle agit.

D'un point de vue syntaxique, en revanche, sur cet exemple, on a pré-cible  $\neq$  cible. Plus précisément, pré-cible  $\subsetneq$  cible. Comme l'élaboration enrichit la description du problème sur laquelle elle agit, on a bien toujours une inclusion (mais, on verra plus loin que cette inclusion n'est pas nécessairement stricte) :

$$pb \subseteq \text{Élaboration}(pb) \quad (5)$$

Autrement dit, Élaboration est *extensive* sur  $(2^{\text{Descripteurs}}, \subseteq)$ .

Que se passe-t-il si on applique Élaboration à cible ? Comme cible est déjà le fruit d'une élaboration, c'est un problème sous forme (déjà) préparée à la remémoration et l'élaboration ne permet donc pas de l'enrichir. Dans ces conditions,  $\text{Élaboration}(\text{cible}) = \text{cible}$  et, en substituant pré-cible par pb, on obtient :

$$\text{Élaboration}(\text{Élaboration}(pb)) = \text{Élaboration}(pb) \quad (6)$$

Autrement dit, la fonction d'élaboration est idempotente :

$$\text{Élaboration} \circ \text{Élaboration} = \text{Élaboration}$$

Enfin, supposons qu'on ait un nouveau problème représenté par pré-cible' et tel que pré-cible  $\subseteq$  pré-cible'. Dans l'exemple des voitures, on peut prendre par exemple :

$$\text{pré-cible}' = \text{pré-cible} \wedge (\text{nb-accidents} = 0)$$

Soit cible' = Élaboration(pré-cible'). Il nous semble raisonnable de considérer que cible' contienne tous les descripteurs de cible (et éventuellement d'autres) : si  $d \in \text{cible} \setminus \text{pré-cible}$  a été inféré à partir de  $Th$  et de descripteurs de pré-cible, alors  $d$  peut également être inféré de descripteurs de pré-cible' et de  $Th$ , pour peu qu'on fasse l'hypothèse d'une inférence monotone (au sens où ce qui a été inféré n'est pas remis en cause par l'ajout de nouvelles informations) et donc il n'est pas incohérent de penser que  $d \in \text{cible}'$ . Donc cible  $\subseteq$  cible'. En substituant pré-cible et pré-cible' par  $pb, pb' \in \text{Problèmes}$ , on a alors :

$$\text{si } pb \subseteq pb' \quad \text{alors } \text{Élaboration}(pb) \subseteq \text{Élaboration}(pb') \quad (7)$$

Autrement dit, Élaboration est une application *croissante* de  $(2^{\text{Descripteurs}}, \subseteq)$  dans  $(2^{\text{Descripteurs}}, \subseteq)$ .

## 4.2. Une définition de l'élaboration

La généralisation de cet exemple nous amène à une définition de l'élaboration :

**Définition 1** Pour des problèmes représentés sous forme d'ensemble de descripteurs —  $\text{Problèmes} = 2^{\text{Descripteurs}}$  — une élaboration est une fonction Élaboration de  $2^{\text{Descripteurs}}$  dans  $2^{\text{Descripteurs}}$  telle que :

$$(i) \quad pb \stackrel{Th}{\equiv} \text{Élaboration}(pb) \text{ pour tout } pb \in 2^{\text{Descripteurs}} \text{ et}$$

(ii) *Élaboration* est une application de fermeture de  $(2^{\text{Descripteurs}}, \subseteq)$ , c'est-à-dire qu'elle est extensive (5), idempotente (6) et croissante (7).

### 4.3. Deux cas particuliers extrêmes

Selon cette définition, il peut exister plusieurs fonctions d'élaboration, parmi lesquelles les deux cas particuliers ci-dessous :

**Cas particulier 1 (identité et élaboration)** L'application *id* définie par  $id(pb) = pb$  pour  $pb \in \text{Problèmes}$  est une fonction d'élaboration.

**Cas particulier 2 (fermeture déductive et élaboration)** Étant donné  $pb \in 2^{\text{Descripteurs}}$ , on appelle *fermeture déductive* de  $pb$  (étant donné la théorie du domaine  $Th$ ), le problème

$$FD_{Th}(pb) = \{d \in \text{Descripteurs} \mid pb \vDash_{Th} d\}$$

où  $pb \vDash_{Th} d$  est vraie si  $d$  est déductible de  $pb$  et de  $Th$ . L'application  $FD_{Th}$  est une fonction d'élaboration.

Le système ÉDHIBOU, qui réalise la saisie et l'élaboration du problème donné en entrée du système RÀPC de KASIMIR 4 (d'Aquin, 2005) effectue une élaboration similaire à  $FD_{Th}$ .

L'application  $FD_{Th}$  nous semble une fonction d'élaboration intéressante. Cependant, elle peut avoir un inconvénient majeur : elle peut donner des problèmes contenant une infinité de descripteurs, par exemple :

- Si  $\text{Descripteurs}$  contient tous les descripteurs ( $\hat{\text{âge}} \leq x$ ) pour  $x \in \mathbb{N}$
- et si  $Th$  permet de faire l'inférence  $(\hat{\text{âge}} \leq x) \vDash_{Th} (\hat{\text{âge}} \leq y)$  pour tout  $x, y \in \mathbb{N}$  tels que  $x \leq y$
- alors  $FD_{Th}(\{\hat{\text{âge}} \leq 10\})$  est infini : il contient tous les descripteurs  $(\hat{\text{âge}} \leq y)$  pour  $y \geq 10$ .

Ces deux cas particuliers de fonctions d'élaboration sont les cas « extrêmes » de fonctions d'élaboration au sens où on peut montrer que, si *Élaboration* vérifie la définition 1, on a :

$$id(pb) \subseteq \text{Élaboration}(pb) \subseteq FD_{Th}(pb)$$

pour tout  $pb \in 2^{\text{Descripteurs}}$  (1).

Pour affiner le choix de la fonction d'élaboration à implanter, la question qui se pose est « Pourquoi élaborer ? » Les deux cas extrêmes introduits ci-dessus peuvent s'interpréter comme deux réponses à cette question :

**(a) Élaboration = id :** Il ne faut pas faire d'élaboration : c'est le problème tel qu'il est décrit par l'utilisateur qui doit être exploité.

---

1. **Preuve :** Comme *Élaboration* est extensive, on a  $pb \subseteq \text{Élaboration}(pb)$  et donc  $id(pb) \subseteq \text{Élaboration}(pb)$ . Soit  $d \in \text{Élaboration}(pb)$ . Étant donné que  $pb \vDash_{Th} \text{Élaboration}(pb)$  (conséquence immédiate de (i)) et que  $\text{Élaboration}(pb) \vDash d$  (puisque *Élaboration*( $pb$ ) est une conjonction de descripteurs et que  $d$  est un de ces descripteurs), on a donc  $pb \vDash_{Th} d$ , d'où  $d \in FD_{Th}(pb)$ . Par conséquent,  $\text{Élaboration}(pb) \subseteq FD_{Th}(pb)$ .  $\square$

(b)  $\text{Élaboration} = \text{FD}_{\text{Th}}$  : Il faut élaborer parce que tout ce qui peut être déduit de la description de l'utilisateur et de la théorie du domaine est susceptible d'être utile pour résoudre ce problème.

Un choix intermédiaire consiste à garder les descripteurs de pré-cible et à y ajouter les descripteurs susceptibles d'être *utiles* à la résolution de problème. Comme celle-ci est effectuée par les deux étapes de remémoration et d'adaptation, l'utilité qui préside au choix de ces descripteurs doit être liée à ces deux étapes et aux connaissances qui les lient. Une façon de construire une fonction d'élaboration selon ce principe est décrite dans la section suivante.

#### 4.4. *Élaboration et dépendances*

Une des approches de l'adaptation s'appuie sur la notion de *dépendance*. Intuitivement, une dépendance est un lien entre un élément de problème et un élément de solution qui indique que le second varie quand le premier varie et comment ces variations sont liées (fonction d'influence). Nous la décrirons, en accord avec les principes proposés dans (Fuchs *et al.*, 2000), par un triplet  $(d, D, \text{Inf})$  où  $d$  est un descripteur de problème,  $D$  est un descripteur de solution et  $\text{Inf}$  est la fonction d'influence.  $\text{Inf}$  est utile à la fonction d'adaptation, mais nous n'en aurons pas besoin pour définir une fonction d'élaboration, aussi noterons-nous ici cette dépendance par un couple  $(d, D)$ .

En toute généralité, une dépendance est fonction du cas auquel elle est attachée ( $D$  peut dépendre de  $d$  dans une partie de l'espace des cas mais ne pas en dépendre dans une autre partie de cet espace). Nous supposons pour simplifier que ce n'est pas le cas et que nous disposons d'un ensemble de dépendances, noté  $\text{Dépendances}$ , indépendant d'un cas particulier.

Par ailleurs, conformément au principe auquel nous adhérons de la remémoration guidée par l'adaptabilité (Smyth *et al.*, 1996), la remémoration doit s'appuyer sur les capacités d'adaptation du système de RÀPC considéré, i.e., en l'occurrence, sur les dépendances connues : parmi les descripteurs du problème cible, ceux dont dépendent des descripteurs de solutions doivent jouer un rôle important pour la remémoration. Aussi, pour préparer une telle remémoration, l'élaboration doit faire ressortir de tels descripteurs et c'est sur cette base que nous proposons la fonction d'élaboration suivante :

$$\text{Élaboration}_{\text{Dépendances}}^{\text{Th}}(\text{pb}) = \text{pb} \cup \{d \in \text{Descripteurs} \mid \text{pb} \models_{\text{Th}} d \text{ et } \exists (d, D) \in \text{Dépendances}\}$$

Autrement dit, l'élaboration ajoute aux descripteurs du problème ceux qui peuvent en être déduits et qui participent aux dépendances déclarées dans l'ensemble  $\text{Dépendances}$ . Par exemple, si  $\text{pb} = d_1 \wedge d_2$ ,  $d_1 \wedge d_2 \models_{\text{Th}} d_3$  et  $(d_3, D_1) \in \text{Dépendances}$ , alors  $d_3 \in \text{Élaboration}_{\text{Dépendances}}^{\text{Th}}(\text{pb})$ . En revanche, si  $\text{pb} = d_1 \wedge d_2$ ,  $d_1 \wedge d_2 \models_{\text{Th}} d_4$  et qu'il n'existe pas de  $(d_4, \cdot) \in \text{Dépendances}$ , alors  $d_4 \notin \text{Élaboration}_{\text{Dépendances}}^{\text{Th}}(\text{pb})$ .

*Remarque.* Notons qu'on peut se demander pourquoi les descripteurs de  $\text{pb}$  qui ne sont pas liés à des dépendances sont conservés par la fonction d'élaboration. En effet, pourquoi les garder puisque l'adaptation ne s'appuie pas sur eux ? Nous proposons deux réponses à cette question : une mauvaise et une bonne. La mauvaise est que si on ne rajoute pas de tels descripteurs, alors la fonction d'élaboration ne respectera pas la définition 1 (en particulier, la propriété d'extensivité). La bonne (selon nous) est que même si ces descripteurs ne jouent pas un rôle connu dans le lien entre problème et solution, ils peuvent jouer un rôle secondaire. Par exemple, supposons

que pré-cible représente une voiture rouge et que  $srce_1$  et  $srce_2$  sont deux cas sources identiques à l'exception de la couleur,  $srce_1$  étant rouge et  $srce_2$  étant bleu métallisé, et qu'on a considéré que la couleur ne joue pas de rôle dans la dépendance. Lequel des deux problèmes sources sera le plus proche de pré-cible? Comme aucun descripteur hors la couleur ne permettra de faire de discrimination, c'est ce descripteur qui sera utilisé : il aurait été dommage de ne plus l'avoir. En l'occurrence, la couleur n'apparaît pas dans les dépendances *connues*, mais rien ne dit qu'elle n'influe pas sur le prix ; en revanche, le fait que l'utilisateur ait fait l'effort de la décrire donne à penser qu'elle peut être pertinente (et donc intervenir dans une dépendance inconnue) : s'il n'a pas décrit le prénom de la grand-mère du premier propriétaire de la voiture, c'est parce qu'il estime que ce descripteur ne peut jouer aucun rôle (à moins que l'interface ne lui ait pas permis de le faire?).

#### 4.5. *Élaboration et mise sous forme canonique*

Une autre piste envisagée pour définir une fonction d'élaboration consiste à utiliser une mise sous forme canonique. Choisir une forme canonique consiste à choisir une convention de représentation qui facilite les comparaisons. Si, par exemple, on veut comparer deux longueurs, on va d'abord les exprimer dans la même unité, puis effectuer la comparaison. En l'occurrence, mettre un problème décrit par un ensemble de descripteurs sous forme canonique c'est le mettre sous une forme qui ne dépend pas de sa « syntaxe d'origine », mais uniquement de sa sémantique. Une fonction d'élaboration qui effectue une mise sous forme canonique aurait la propriété suivante, pour  $pb, pb' \in \text{Problèmes}$  :

$$pb \stackrel{\text{Th}}{\equiv} pb' \quad \Rightarrow \quad \text{Élaboration}(pb) = \text{Élaboration}(pb')$$

(où  $\equiv$  dénote ici l'égalité entre ensembles, i.e., modulo l'ordre des éléments).

Malheureusement, une fonction d'élaboration vérifiant cette implication et la définition 1 n'est pas nouvelle : elle coïncide nécessairement avec  $FD_{\text{Th}}$  (2).

#### 4.6. *Élaboration et résolution partielle de problème*

Une autre façon de considérer l'élaboration est de voir dans cette étape du RÀPC une première tentative de résolution directe de problème : on s'appuie sur  $\text{Th}$  pour déduire ce qu'on peut de pré-cible, dans l'optique de sa résolution.  $\text{Th}$  peut éventuellement permettre de déduire des informations sur ce qui relève de la solution cherchée du problème, par exemple une contrainte qui va restreindre l'ensemble des solutions admissibles.

À titre d'exemple, supposons que la connaissance de l'Argus soit disponible et fasse partie de la théorie du domaine  $\text{Th}$ , mais qu'on n'accorde qu'une confiance limitée sur sa précision. Plus précisément, on est capable d'avoir une estimation  $e_{\text{prix}}$  du prix d'une voiture en fonction de différentes caractéristiques, mais on estime que le prix réel  $\text{prix}$  doit être compris entre la moitié et le double de l'estimation :

$$\frac{e_{\text{prix}}}{2} \leq \text{prix} \leq 2e_{\text{prix}}$$

---

2. **Preuve :** Soit  $pb \in \text{Problèmes}$  et  $pb' = FD_{\text{Th}}(pb)$ .  $pb \stackrel{\text{Th}}{\equiv} pb'$ , par définition de  $FD_{\text{Th}}$ . Par conséquent,  $\text{Élaboration}(pb) = \text{Élaboration}(pb')$ . Or,  $\text{Élaboration}$  est extensive, donc  $FD_{\text{Th}}(pb) = pb' \subseteq \text{Élaboration}(pb)$ . Comme  $FD_{\text{Th}}$  majore toute fonction d'élaboration on a aussi  $\text{Élaboration}(pb) \subseteq FD_{\text{Th}}(pb)$ . Par conséquent,  $\text{Élaboration}(pb) = FD_{\text{Th}}(pb)$ .  $\square$

Des connaissances de l'Argus et de la connaissance de cet encadrement, on peut inférer  $e_{\text{prix}}$  et donner une contrainte pour prix :

Si pré-cible  $\models_{\text{Th}} (e_{\text{prix}} = 6000)$  alors pré-cible  $\models_{\text{Th}} (3000 \leq \text{prix} \leq 12000)$

Supposons que la fonction d'élaboration choisie retienne ces deux descripteurs ( $e_{\text{prix}} = 6000$ ) et  $(3000 \leq \text{prix} \leq 12000)$ . Cela permettra alors d'invalider toute solution  $\text{Sol}(\text{cible})$  qui ne respecterait pas cette contrainte.

Notons en l'occurrence que cette contrainte peut être considérée comme une solution partielle (et donc un descripteur solution) mais qu'on peut également la considérer comme un descripteur de problème, vu que le problème après élaboration devient « Estimer le prix d'une voiture ayant les caractéristiques suivantes: [...], sachant que ce prix doit être compris entre 3000 et 12000 euros. » (la séparation dans un cas entre sa partie problème et sa partie solution n'est pas toujours évidente...).

## 5. Discussion : élaboration et principe d'indépendance à la syntaxe

Le principe d'indépendance à la syntaxe est une propriété généralement admise pour un raisonnement. Il stipule que si deux formules  $f$  et  $g$  sont logiquement équivalentes dans le cadre d'une théorie  $\text{Th}$  ( $f \stackrel{\text{Th}}{\equiv} g$ ), alors ce qu'on peut inférer de  $f$  sera également inférable de  $g$ .

Par exemple, le RÀPC est un raisonnement et nous supposons qu'il vérifie ce principe. De plus, nous supposons qu'il est fonctionnel, au sens où étant donné pré-cible,  $\text{Th}$  et  $\text{BC}$  ( $\text{BC}$  étant la base de cas), un système de RÀPC donné fournira toujours le même résultat  $\text{Sol}(\text{cible})$ . Dans ces conditions, le principe d'indépendance à la syntaxe entraîne que

$$\text{pré-cible} \stackrel{\text{Th}}{\equiv} \text{pré-cible}' \quad \Rightarrow \quad \text{RàPC}(\text{pré-cible}, \text{Th}, \text{BC}) \stackrel{\text{Th}}{\equiv} \text{RàPC}(\text{pré-cible}', \text{Th}, \text{BC})$$

Autrement dit, un système de RÀPC ne doit pas être sensible à la façon dont le problème en entrée est saisi. Par exemple, si  $\text{pré-cible} = d_1 \wedge d_2 \wedge d_3$ ,  $\text{pré-cible}' = d_2 \wedge d_1$  et  $d_1 \models_{\text{Th}} d_3$ , alors  $\text{pré-cible}$  et  $\text{pré-cible}'$  représentent le même problème, et la résolution de l'un doit donner un résultat équivalent à la résolution de l'autre.

Le RÀPC est composé des trois étapes d'élaboration, de remémoration et d'adaptation (nous laissons de côté la mémorisation qui ne participe pas à la résolution du problème courant) :

Élaboration :  $\text{pré-cible} \mapsto \text{cible}$

Remémoration :  $(\text{cible}, \text{BC}) \mapsto (\text{srce}, \text{Sol}(\text{srce}))$

Adaptation :  $((\text{srce}, \text{Sol}(\text{srce})), \text{Sol}(\text{cible})) \mapsto \text{Sol}(\text{cible})$

Notons que  $\text{Th}$  est un paramètre sous-entendu de ces trois fonctions. Rien n'oblige chacune de ces étapes à respecter (individuellement) le principe d'indépendance à la syntaxe, mais il doit être vérifié globalement. Cependant, une fonction d'élaboration respectant la définition 1 le vérifiera :

$$\begin{aligned} \text{pré-cible} \stackrel{\text{Th}}{\equiv} \text{pré-cible}' \\ \Rightarrow \text{Élaboration}(\text{pré-cible}) \stackrel{\text{Th}}{\equiv} \text{Élaboration}(\text{pré-cible}') \end{aligned}$$

pour tout pré-cible, pré-cible'  $\in$  Problèmes <sup>(3)</sup>.

Par ailleurs, nous supposons que la base de cas ne contient pas de cas redondants modulo la théorie Th :

$$\begin{aligned} (\text{srce}_1, \text{Sol}(\text{srce}_1)) &\stackrel{\text{Th}}{\equiv} (\text{srce}_2, \text{Sol}(\text{srce}_2)) \\ &\Rightarrow (\text{srce}_1, \text{Sol}(\text{srce}_1)) = (\text{srce}_2, \text{Sol}(\text{srce}_2)) \end{aligned}$$

pour  $(\text{srce}_1, \text{Sol}(\text{srce}_1)), (\text{srce}_2, \text{Sol}(\text{srce}_2)) \in \text{BC}$ . De plus, il nous semble raisonnable de supposer que le choix d'un cas source ne dépend pas de la forme du problème saisi par l'utilisateur. Dans ce cas, on a l'implication suivante :

$$\begin{cases} \text{pré-cible} \stackrel{\text{Th}}{\equiv} \text{pré-cible}' \\ \text{cible} = \text{Élaboration}(\text{pré-cible}) \\ \text{cible}' = \text{Élaboration}(\text{pré-cible}') \end{cases} \Rightarrow \text{Remémoration}(\text{cible}, \text{BC}) = \text{Remémoration}(\text{cible}', \text{BC}) \quad (8)$$

Pour pousser plus loin notre discussion, nous considérerons deux cas.

*Premier cas :* la remémoration ne vérifie pas le principe d'indépendance à la syntaxe étant donné Th. Dans ce cas, (8) est une propriété supplémentaire que devrait vérifier la fonction d'élaboration pour préparer la remémoration (sous les différentes hypothèses introduites ci-dessus). En particulier, si Élaboration est une mise sous forme canonique (et donc, comme on l'a vu à la section 4.5,  $\text{Élaboration} = \text{FD}_{\text{Th}}$ ), ce sera vérifié. De façon générale, cette condition (8) peut s'interpréter par « L'élaboration associe à un problème pré-cible  $\in$  Problèmes un problème cible qui appartient à un sous-ensemble de Problèmes tel que la remémoration respecte l'indépendance à la syntaxe pour les problèmes de ce sous-ensemble. »

*Deuxième cas :* la remémoration vérifie le principe d'indépendance à la syntaxe étant donnée Th. Dans ce cas, on a :

$$\text{cible} \stackrel{\text{Th}}{\equiv} \text{cible}' \Rightarrow \text{Remémoration}(\text{cible}, \text{BC}) = \text{Remémoration}(\text{cible}', \text{BC})$$

pour tout cible, cible'  $\in$  Problèmes. Dans ce cas, quelle que soit la fonction d'élaboration choisie respectant la définition 1 (y compris id), la remémoration donnera le même résultat. Par conséquent, dans ce cas, l'élaboration en tant que telle est inutile. Néanmoins, l'étude de l'élaboration, comme elle a pour objet de faire ressortir les descripteurs utiles à la remémoration (c'est particulièrement vrai pour l'élaboration s'appuyant sur les dépendances, cf. section 4.4), permet de mieux pondérer ces descripteurs, ce qui est utile à la conception de la fonction de remémoration.

## 6. Conclusion et perspectives

Cet article tente une première formalisation de l'étape d'élaboration du RÀPC, dont le rôle est de préparer les étapes suivantes : la remémoration, qui la suit immédiatement, mais aussi l'adapt-

---

3. **Preuve :** Supposons que pré-cible  $\stackrel{\text{Th}}{\equiv}$  pré-cible'. D'après (i) de la définition 1, on a pré-cible  $\stackrel{\text{Th}}{\equiv}$  Élaboration(pré-cible) et pré-cible'  $\stackrel{\text{Th}}{\equiv}$  Élaboration(pré-cible'). Comme  $\stackrel{\text{Th}}{\equiv}$  est une relation symétrique et transitive, on peut conclure : Élaboration(pré-cible)  $\stackrel{\text{Th}}{\equiv}$  Élaboration(pré-cible').  $\square$

tation si on considère, comme nous le faisons, que le lien entre remémoration et adaptation est fondamental en RÀPC. Dans certaines situations, implanter effectivement l'élaboration est inutile : c'est le cas si chaque étape du RÀPC est supposée respecter le principe d'indépendance à la syntaxe modulo toutes les connaissances du domaine disponibles (Th, dans cet article). Dans ce cas, l'élaboration est en quelque sorte incluse dans ces étapes. Même alors, étudier l'élaboration nous semble profitable : cette étude permet de faire émerger des liens entre la représentation des cas et la représentation des connaissances générales utiles à la remémoration et à l'adaptation. En particulier, comme on l'a vu dans la section 4.4, étudier l'élaboration peut revenir à l'étude des dépendances entre problèmes et solutions, lesquelles dépendances participent à la fois à la représentation des cas, à la remémoration (la pertinence des descripteurs problèmes pour la remémoration est fonction des dépendances) et à l'adaptation.

Cette première formalisation de l'élaboration est partie d'un exemple très simple. Elle doit se poursuivre par sa confrontation à d'autres exemples, plus réalistes, même si l'exemple choisi est issu d'une réflexion sur de tels exemples. La réflexion, encore préliminaire sur l'élaboration et le principe d'indépendance à la syntaxe doit être poursuivie. Enfin, le lien avec l'élaboration et l'indexation — qui peut également être considérée comme une étape préliminaire de la remémoration — doit être étudié<sup>4</sup>.

## 7. Références

- Aarts R. J. et Rousu J. (1996). « Towards CBR for Bioprocess Planning. », dans I. SMITH et B. FALTINGS (éds), *EWCBR*, LNAI 1168, pp. 16–27.
- Bento C. et Costa E. (1993). « A Similarity Metric for Retrieval of Cases Imperfectly Explained. », dans M. RICHTER, S. WESS, K.-D. ALTHOFF et F. MAURER (éds), *Proceedings of the First European Workshop on Case-Based Reasoning (EWCBR'93)*, Kaiserslautern, pp. 92–105.
- Cain T. , Pazzani M. J. et Silverstein G. (1991). « Using Domain Knowledge to Influence Similarity Judgements », *Proceedings of the Case-Based Reasoning Workshop*, Washington D.C., U.S.A., pp. 191–198.
- d'Aquin M. (2005). *Un portail sémantique pour la gestion des connaissances en cancérologie*, Thèse d'université, Université Henri Poincaré Nancy 1, soutenue le 15 décembre 2005.
- Fuchs B. , Lieber J. , Mille A. et Napoli A. (2000). « An Algorithm for Adaptation in Case-Based Reasoning », *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000)*, Berlin, Germany, pp. 45–49.
- Goel A. , Bhatta S. et Stroulia E. (1997). *Issues and Applications of Case-Based Reasoning in Design*, Erlbaum, chapter Kritik: An Early Case-Based Design System.
- Herbeaux O. (2000). *ACCELERE : aide à la conception de caoutchouc cellulaire exploitant la remémoration d'expériences*, Thèse d'université, Université Jean Monnet de Saint-Étienne.
- Riesbeck C. K. et Schank R. C. (1989). *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey.
- Rousu J. et Aarts R. J. (1996). « Adaptation Cost as a Criterion for Solution Evaluation », *EWCBR*, pp. 354–361.
- Smyth B. et Keane M. T. (1996). « Using adaptation knowledge to retrieve and adapt design cases », *Knowledge-Based Systems*, 9(2): 127–135.
- RÀPC(2005). *Actes du treizième atelier raisonnement à partir de cas, RàPC'05*, Sylvie Després (éditeur scientifique) plate-forme AFIA.

---

4. **Remerciements** : Les auteurs tiennent à remercier les lecteurs de cet article pour leurs remarques encourageantes et constructives. Certaines de ces remarques n'ont pas été complètement intégrées dans cet article mais pourront s'avérer utile à des réflexions et discussions ultérieures.