



HAL
open science

Vers une théorie unifiée de l'adaptation en raisonnement à partir de cas

Béatrice Fuchs, Jean Lieber, Alain Mille, Amedeo Napoli

► **To cite this version:**

Béatrice Fuchs, Jean Lieber, Alain Mille, Amedeo Napoli. Vers une théorie unifiée de l'adaptation en raisonnement à partir de cas. Journées Ingénierie des Connaissances - IC'99, 1999, Palaiseau, France. inria-00098764

HAL Id: inria-00098764

<https://inria.hal.science/inria-00098764v1>

Submitted on 26 Sep 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 théorie unifiée de l'adaptation en raisonnement à partir de cas

Béatrice Fuchs¹, Jean Lieber², Alain Mille³, Amedeo Napoli²

¹ Université Lyon 3, IAE - MODEME, 15 quai Claude Bernard, 69239 Lyon cedex 02,

² LORIA - UMR 7503, BP 239, 54506 Vandoeuvre-les-Nancy cedex

³ LISA - CPE - Lyon, 43 bd du 11 novembre 1918, 69 616 Villeurbanne cedex

e-mail : fuchs@univ-lyon3.fr, lieber@loria.fr, mille@cpe.fr, napoli@loria.fr

Résumé

Le raisonnement à partir de cas (RàPC) exploite des épisodes de résolution de problème mémorisés appelés cas afin de résoudre de nouveaux problèmes. L'étape d'adaptation, la plus complexe et la plus importante en RàPC, est généralement étudiée dans le cadre restreint d'un domaine d'application. Dans cet article, nous proposons une première analyse de l'adaptation de cas indépendamment d'un domaine d'application spécifique. Nous proposons de combiner les étapes de remémoration et d'adaptation en un processus unique de planification qui construit une suite ordonnée d'opérations pour passer d'un état initial (le problème posé) à un état final (le problème résolu). Pour cela, nous argumentons l'hypothèse qu'un cas – décrivant un problème et sa solution – peut être considéré comme un plan de résolution de problème. Ainsi, le problème de l'adaptation d'un cas peut être appréhendé en étudiant le problème de l'adaptation d'un plan. Enfin, il est montré comment la remémoration et l'adaptation d'un cas peuvent être associées grâce aux notions de reformulation et de chemin de similarité.

1 Introduction

Le raisonnement à partir de cas (RàPC) associe à un problème P une solution construite en réutilisant la solution mémorisée d'un problème P' similaire à P . Le cycle de RàPC est composé de trois étapes principales : la remémoration consiste à rechercher un problème similaire P' dans une base de cas ; l'adaptation consiste à adapter la solution du problème similaire P' ; et la mémorisation éventuelle du problème P et de sa solution dans la perspective de sa réutilisation future termine le cycle. L'implantation de ce cycle de raisonnement a donné naissance aux systèmes de RàPC qui exploitent une base de cas couplée éventuellement avec d'autres bases de connaissances afin de résoudre des problèmes

de conception, d'interprétation, de diagnostic, de planification, etc.

Une des idées maîtresses du RàPC est d'*adapter* la solution d'un problème connu afin de résoudre un nouveau problème. Beaucoup de chercheurs dans le domaine du RàPC pensent que l'adaptation est difficile à modéliser, qu'elle dépend principalement du domaine d'application et que, par conséquent, elle doit être implantée de façon *ad hoc*.

Dans cet article, nous proposons un modèle général de l'adaptation qui s'appuie sur les principes d'adaptation d'un plan (cette approche est également étudiée dans [9]). De plus, l'adaptation elle-même peut être considérée comme un processus de planification dont l'état initial est la solution de départ, (la solution du problème connu P'), et l'état final la solution adaptée (la solution du problème P). Dans la suite de l'article, un plan est considéré comme un triplet (I, B, O) , où I est un état initial, B un but et O un ensemble d'opérations permettant de satisfaire le but à partir de données associées à I .

Une telle modélisation de l'adaptation permet de concevoir, en théorie et en pratique, la combinaison de l'adaptation et de la remémoration, afin d'assurer la remémoration d'un cas adaptable et de guider l'adaptation d'un tel cas.

L'approche présentée dans cet article est issue d'un travail en cours sur la formalisation du RàPC que nous souhaitons réutiliser pour concevoir des systèmes de RàPC dans différents domaines.

Le problème de l'adaptation est détaillé dans le paragraphe 2, et quelques approches significatives de l'adaptation sont résumées en paragraphe 3. Puis nous présentons notre approche en deux étapes : le paragraphe 4 montre comment un cas peut être considéré comme une forme simplifiée de plan, puis le paragraphe 5 explique comment la remémoration et l'adaptation peuvent être appréhendées dans un cadre unifié. Un exemple complet de cette approche est présenté dans le paragraphe 6. Nous terminons par une discussion et une conclusion (paragraphe 7 et 8).

2 Le problème de l'adaptation en RàPC

Le RàPC a pour objectif de résoudre un problème appelé *problème cible* (ou nouveau problème) et noté *cible* en utilisant une *base de cas* constituée d'un ensemble fini de cas. Un cas noté $(P, \text{Sol}(P))$ est défini par un problème P et sa solution $\text{Sol}(P)$. Un cas de la base de cas est appelé *cas source* et noté $(\text{source}, \text{Sol}(\text{source}))$ où *source* est appelé le *problème source*. La remémoration consiste à choisir dans la base de cas un cas $(\text{source}, \text{Sol}(\text{source}))$ *similaire* à *cible*¹. L'*adaptation* consiste à s'inspirer de la solution $\text{Sol}(\text{source})$ afin de résoudre *cible*. Elle peut être symbolisée par la figure 1 qu'il convient de lire de la manière suivante: « Étant donné *source*, $\text{Sol}(\text{source})$, *cible* et les relations entre ces objets, quelle est $\text{Sol}(\text{cible})$? » Ce schéma de principe du RàPC est inspiré de [4].

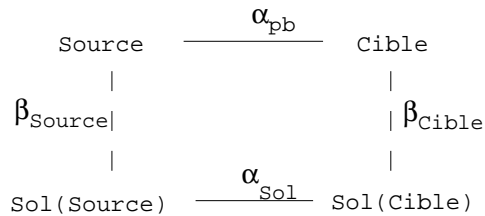


FIG. 1 – Le schéma de principe du RàPC. Les relations β représentent les relations entre problèmes et solutions; les relations α représentent les relations entre les sources et cibles.

Le RàPC repose sur la réutilisation de l'expérience passée pour résoudre un nouveau problème. Par définition, la réutilisation implique une reconsidération de l'expérience passée dans un nouveau contexte, ce qui nécessite son adaptation à ce nouveau contexte. Ainsi, l'objectif de la remémoration d'un cas similaire est de trouver un cas adaptable. Il est donc important que la connaissance utilisée pour la remémoration soit directement reliée à la connaissance utilisée pour l'adaptation. En d'autres termes, le concepteur d'un système de RàPC ne doit pas, en pratique, dissocier ces deux sources de connaissance. Même si la tâche d'adaptation est réalisée par l'utilisateur ou un autre système de résolution de problème (dans ce cas, elle sera considérée comme un sous-problème du cycle de raisonnement principal), les connaissances contrôlant l'étape de remémoration devraient être étroitement reliées à celles utilisées par l'adaptation. Idéalement, le cas remémoré doit être adaptable afin que sa réutilisation soit garantie pour résoudre un nouveau problème.

1. En fait, la similarité est généralement évaluée entre *source* et *cible*, mais $\text{Sol}(\text{source})$ peut jouer un rôle dans cette évaluation. En effet, la partie de *source* la plus pertinente en accord avec sa solution $\text{Sol}(\text{source})$ peut être utilisée pour évaluer la similarité (voir par exemple [34]).

Considérons de nouveau le schéma de principe du RàPC de la figure 1. Il est possible de le considérer selon deux dimensions. La dimension verticale correspondant aux relations β exprime les relations entre les problèmes et les solutions et est précisée dans le paragraphe 4. La dimension horizontale correspondant aux relations α représente les relations entre les cas source et cible et est précisée dans le paragraphe 5: la relation α_{pb} , orientée de *cible* vers *source*, correspond au processus de remémoration; la relation α_{Sol} , orientée de $\text{Sol}(\text{source})$ vers $\text{Sol}(\text{cible})$, correspond au processus d'adaptation. Ainsi, la remémoration et l'adaptation peuvent être vues comme deux processus symétriques.

Avant de présenter ces deux vues, nous présentons dans le paragraphe suivant d'autres approches de l'adaptation.

3 Quelques approches de l'adaptation

Parmi les travaux récents sur l'adaptation, ceux présentés ci-dessous sont ceux qui nous ont le plus inspirés dans nos travaux et qui ont le plus marqué la recherche sur l'adaptation (voir également [36]).

- L'analogie par *dérivation* et l'analogie par *transformation* ont été introduites dans [4] et [34] comme deux processus d'adaptation différents. La première consiste à adapter le processus de construction de la solution mémorisée afin de produire la solution à un nouveau problème, alors que la seconde consiste à recopier la solution mémorisée et à substituer certains éléments par d'autres afin de produire la solution au nouveau problème.
- La remémoration guidée par l'adaptation décrite dans [33] est une approche de la remémoration prenant en compte des connaissances d'adaptation. Ces connaissances sont représentées par des *spécialistes d'adaptation* et des *stratégies d'adaptation*. Lorsqu'un problème source est comparé à un problème cible, les spécialistes nécessaires à l'adaptation de ce cas sont mis en évidence. Le cas nécessitant le moins d'effort d'adaptation est choisi et ensuite adapté par l'application des spécialistes guidée par les stratégies. Les stratégies contrôlent en particulier l'ordre d'application des spécialistes.
- Dans [20] est présentée une approche de l'adaptation qui repose sur une recherche en

mémoire des meilleurs cas dans lesquels certains éléments de solution seront substitués. Cette recherche est basée sur une similarité contextuelle (voir aussi [8], [17], [18], [19]).

- Le problème de l'adaptation peut être vu comme un problème de satisfaction de contraintes, ce qui suppose qu'une modélisation sous forme de contraintes soit adaptée pour le type de problème étudié (ce type de modèle est bien adapté pour les problèmes de simulation, de conception, d'architecture, etc.). La solution d'un nouveau problème est obtenue par satisfaction des nouvelles contraintes et en transformant une solution mémorisée [11] (voir également [12], [31], [30], [15]).
- Dans le cadre de l'adaptation hiérarchique [2], des cas abstraits (c'est à dire des cas permettant de considérer un problème à différents niveaux d'abstraction: plus le contexte est abstrait, plus le problème est résolu « aisément ») sont exploités à la place de cas concrets et les solutions sont *raffinées*, c'est à dire élaborées progressivement dans les différents niveaux d'abstraction, permettant d'obtenir une solution au nouveau problème (voir également [37], [35], [38], [1], [3], [32]).
- Dans [14], la planification à partir de cas est étudiée dans le cadre des logiques de descriptions et analysée d'un point de vue formel: les cas sont des plans représentés par des formules dans une logique temporelle et les index de cas sont représentés par des concepts dans une logique de descriptions. Les index sont organisés en une hiérarchie exploitée par un processus de classification qui constitue le mécanisme de base de la remémoration. Un problème *source* peut être choisi et réutilisé afin de résoudre un problème *cible* si les index associés aux états initiaux et finaux de ces problèmes vérifient les contraintes suivantes: $\text{idx}(\text{init}_{\text{cible}}) \sqsubseteq \text{idx}(\text{init}_{\text{source}})$ et/ou $\text{idx}(\text{but}_{\text{source}}) \sqsubseteq \text{idx}(\text{but}_{\text{cible}})$, où *init* est un état initial, *but* un état final et \sqsubseteq détermine la relation de généralité entre les index d'une hiérarchie. Le connecteur *et* correspond à une remémoration *forte* et le connecteur *ou* à une remémoration *faible*. Si la remémoration est forte, alors tout cas source vérifiant les conditions ci-dessus peut être réutilisé pour résoudre le problème cible. Si la remémoration est faible, le meilleur cas est celui partageant le plus de sous-buts ou préconditions (ou les deux) avec le problème cible.

Dans ces conditions, il est possible de fixer un seuil minimal significatif de l'effort d'adaptation, et si aucun cas candidat ne satisfait ce seuil, la construction d'une solution du problème cible doit être réalisée à l'aide de techniques de planification directe.

Chacun de ces travaux traite un aspect particulier de l'adaptation à un niveau d'abstraction donné. Cependant, le processus d'adaptation est pratiquement toujours considéré indépendamment des autres opérations du RàPC, en particulier la remémoration. Afin d'aller plus loin, il est nécessaire de s'intéresser à deux problèmes: (1) définir d'une manière générale ce qu'est un cas et modéliser l'adaptation indépendamment d'un domaine d'application particulier, (2) combiner les processus de remémoration et d'adaptation en un processus unique. Les travaux précédents, les applications réalisées par Mille, Fuchs et Herbeaux et les travaux de Melis, Lieber et Napoli, nous ont permis de modéliser un point de vue unifié sur l'adaptation et d'apporter des premiers éléments de réponse aux questions (1) et (2):

- Des systèmes tels que DESIGNER [6] pour la conception en supervision industrielle, PAD'IM [7], [26] pour l'aide à la décision, ACCELERE [10] ou SYSEXPERT [25] ont constitué le terrain d'étude pour une première tentative vers une théorie unifiée de l'adaptation [27]. L'adaptation d'un cas est considérée comme l'adaptation d'un plan dont les étapes sont considérées à différents niveaux de granularité, ce qui permet de modéliser l'adaptation à l'aide d'opérations simples telles que l'ajout ou la suppression d'éléments à un même niveau de granularité.
- Les travaux présentés d'une part dans [23], et d'autre part dans [21] et [22] partagent le même point de vue sur l'adaptation, ce qui a abouti aux notions de reformulation et de chemin de similarité [24].

Une *reformulation* est un couple (r_P, r_S) où r_P est une relation entre deux problèmes pb_0 et $\text{pb}_1 - \text{pb}_0$ r_P pb_1 - et r_S est une relation fonctionnelle associant à une solution connue $\text{Sol}(\text{pb}_0)$ de pb_0 une solution $\text{Sol}(\text{pb}_1)$ de $\text{pb}_1 - \text{Sol}(\text{pb}_0) \xrightarrow{r_S} \text{Sol}(\text{pb}_1)$. r_S est appelée *fonction d'adaptation spécifique*. Deux reformulations (r_P^1, r_S^1) et (r_P^2, r_S^2) peuvent se composer: si pb_0 r_P^1 pb_1 et pb_1 r_P^2 pb_2 alors il est possible d'adapter une solution $\text{Sol}(\text{pb}_0)$ de pb_0 en une solution $\text{Sol}(\text{pb}_2)$ de pb_2 en appliquant r_S^1 et r_S^2 en séquence: $\text{Sol}(\text{pb}_0) \xrightarrow{r_S^1} \text{Sol}(\text{pb}_1) \xrightarrow{r_S^2} \text{Sol}(\text{pb}_2)$. Plus généralement, une séquence pb_0 r_P^1 $\text{pb}_1 \dots \text{pb}_{q-1}$ r_P^q pb_q est appelée *chemin de similarité* entre pb_0 et pb_q . La remémoration

consiste à trouver un chemin de similarité entre un problème source (à choisir dans la base de cas) et le problème cible. L'adaptation consiste à appliquer en séquence les fonctions d'adaptation spécifiques r_S^i . Ainsi, la remémoration permet non seulement de trouver un cas adaptable, mais elle construit également un chemin de similarité qui sera exploité par le processus d'adaptation.

Ce qui est présenté ci-après est issu d'un travail de recherche en cours ayant pour objectif d'« unifier » les travaux présentés dans [27], dans lequel un cas est considéré comme un plan, et ceux décrits dans [24], qui combinent la remémoration et l'adaptation en un même processus.

4 Un cas vu comme un plan

Un problème à résoudre est représenté par une description d'un problème cible pour lequel il faut trouver une solution $\text{Sol}(\text{cible})$. La description du problème comprend le but (l'ensemble des descripteurs que la solution doit vérifier) et le contexte initial (l'ensemble des descripteurs qui sont vérifiés au début du processus de résolution de problème). Pour simplifier, un descripteur peut être soit un couple (attribut, valeur), soit une contrainte devant être vérifiée. Vérifier un descripteur signifie soit trouver la valeur de l'attribut correspondant, soit vérifier la contrainte correspondante. La description de la solution est composée d'un ensemble de descripteurs qui doivent être vérifiés *afin d'atteindre le but* étant donné le contexte initial du cas (figure 2). La description du cas peut donc être vue comme un plan.

On peut remarquer que, lorsque le plan de construction de la solution est connu, alors la solution peut être immédiatement calculée, elle est donc également connue. Par conséquent, *le plan de construction de la solution peut être considéré comme étant véritablement la solution* (ou au moins, la solution retournée par le processus de remémoration). En combinant la remémoration et l'adaptation et en considérant le processus de remémoration/adaptation comme une planification qui peut se modéliser sous la forme d'un triplet (I, B, O) , il n'y a pas de différence fondamentale entre la substitution d'éléments de solution et la substitution de certaines étapes.

Conformément à ce point de vue, si un plan est représenté par le triplet (I, B, O) , il est possible de représenter un cas par un état initial – le contexte initial – un état final – le but – et une suite d'opérations d'adaptation O permettant d'atteindre l'état B à partir de l'état I (pour simplifier, on supposera qu'une telle suite est une liste totalement ordonnée de couples état-opérateur).

Cette hypothèse n'est pas trop forte, car l'élaboration de la solution peut être vue comme un plan de résolution de problème, (voir en particulier [29] [28] [16] où ce point de vue est discuté), c'est à dire qu'un état initial, un but et une collection d'opérateurs permettant de générer un espace d'états sont nécessaires (un tel problème est qualifié de *problème de recherche* dans [5]).

L'adaptation d'un cas peut alors se ramener à une planification, ce qui permet d'exploiter les travaux sur la planification indépendante du domaine [9].

Dans [9] deux approches de l'adaptation de plan ont été présentées: une approche générative qui suppose que toutes les connaissances nécessaires pour générer un plan de solution à partir de rien sont disponibles, et une approche par transformation où l'hypothèse précédente n'est pas nécessaire et basée sur la réutilisation (dans ce dernier cas, il n'est pas garanti de trouver une solution). En fait, dans les deux approches, les connaissances nécessaires pour identifier les étapes à transformer expriment d'une manière ou d'une autre le rôle joué par une étape (élément de la solution) afin de vérifier la description du but tout en respectant la description de l'état initial. Les différences entre la description du problème cible et celle du problème source mémorisé permettent d'évaluer l'écart à réduire dans la description de la solution source afin d'élaborer la description du problème cible. Par exemple, si *source* est le problème « Quel est l'itinéraire pour aller de Paris à Munich ? », et *cible* « Quel est l'itinéraire pour aller de Lyon à Munich ? » la différence observée – la différence entre les états initiaux Paris et Lyon – nécessite de modifier le trajet Paris-Munich en un trajet Lyon-Munich, par exemple en concaténant le trajet Lyon-Paris avec le trajet Paris-Munich. Dans l'approche générative, il faut partir d'un plan général qui ne soit pas en contradiction avec le problème, et générer les étapes manquantes, alors que dans l'approche par transformation, il faut identifier les étapes à substituer ou à modifier (une modification simple, ou une séquence de modifications). Dans les deux cas, l'adaptation peut être décomposée en une généralisation du cas source compatible avec le problème cible, puis en une spécialisation vérifiant les descripteurs du problème cible. Le paragraphe suivant détaille les étapes génériques d'une telle adaptation.

5 Adapter un cas comme un plan

Résoudre un problème consiste à *construire la liste (ordonnée ou non) des descripteurs de solution menant à la satisfaction du but*. Dans le domaine de la planification (sans utiliser l'expérience d'une base de cas), il s'agit d'un problème de génération de plan qui a fait l'objet de nombreux travaux. Lorsque l'on a

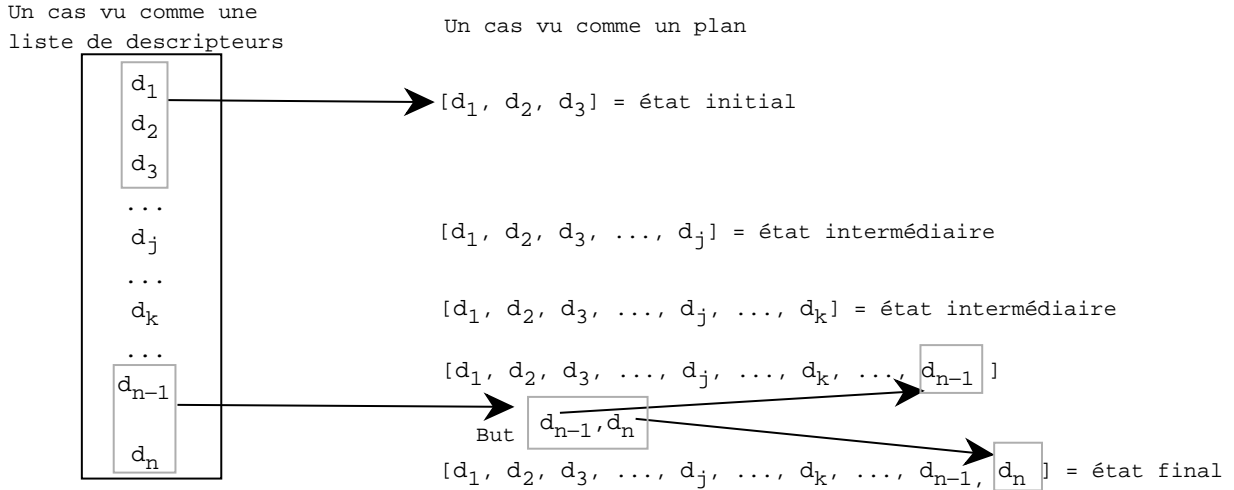


FIG. 2 – Les descripteurs $[d_1, d_2, d_3]$ représentent ce qui est vérifié au début du processus de résolution de problème (l'état initial), et $[d_k, d_{k+1}]$ représentent ce qui doit être vérifié lorsque le problème sera résolu (le but). Les autres descripteurs représentent les autres parties de la solution. À partir de l'état initial, le processus de résolution de problème peut être représenté par ses différents états intermédiaires intégrant de nouveaux éléments de solution jusqu'à ce que le but soit satisfait.

recours à l'expérience concrète, typiquement en RàPC, cela implique une approche différente (voir figure 3, qui constitue une instanciation du schéma de principe du RàPC de la figure 1) :

- Elaborer un index du problème cible, noté $\text{idx}(\text{cible})$. Cet index est constitué des descripteurs pertinents (état initial, but) du problème à résoudre.
- Trouver un index $\text{idx}(\text{source})$ d'un cas source similaire à $\text{idx}(\text{cible})$. Chaque cas de la base de cas doit être associé à un index pour que cette comparaison soit possible et complète.
- À partir de l'index $\text{idx}(\text{source})$, il est possible d'accéder naturellement au problème source.
- La solution $\text{Sol}(\text{source})$ de source est choisie comme base pour résoudre le problème cible. Cette étape fait le lien entre la remémoration dans l'espace des descriptions de problèmes et l'adaptation dans l'espace des descriptions de solutions.
- La solution $\text{Sol}(\text{source})$ est généralisée afin d'être cohérente avec l'index $\text{idx}(\text{source})$: une solution $\text{Sol}(\text{idx}(\text{source}))$ de l'index du problème source est déterminée.
- La solution $\text{Sol}(\text{idx}(\text{source}))$ est spécialisée pour devenir une solution $\text{Sol}(\text{idx}(\text{cible}))$ du problème cible généralisé $\text{idx}(\text{cible})$.

- La solution $\text{Sol}(\text{idx}(\text{source}))$ est spécialisée afin de prendre en compte les simplifications réalisées lors de la généralisation de cible en $\text{idx}(\text{cible})$ (correspondant à l'étape d'élaboration (a)).

On peut remarquer que cette approche est une spécialisation de l'approche plus générale de reformulation présentée au paragraphe 3. Les étapes (a), (b) et (c) correspondent à la construction du chemin de similarité lors de la remémoration, et les étapes (e), (f) et (g) correspondent à l'application de fonctions d'adaptation spécifiques lors de l'adaptation. On peut également remarquer que l'étape (b) et l'étape correspondante (f) peuvent être décomposées en plusieurs sous-étapes. Dans [21], l'étape de *classification dure* (b) consiste à rechercher un index $\text{idx}(\text{source})$ plus général que $\text{idx}(\text{cible})$; dans l'étape de *classification élastique*, il s'agit de rechercher une suite de relations r_P entre un index $\text{idx}(\text{source})$ à choisir et l'index $\text{idx}(\text{cible})$.

6 Un exemple en planification d'itinéraire

Dans cet exemple, il s'agit de déterminer un itinéraire afin d'atteindre une ville d'arrivée à partir d'une ville de départ en utilisant un réseau de routes.

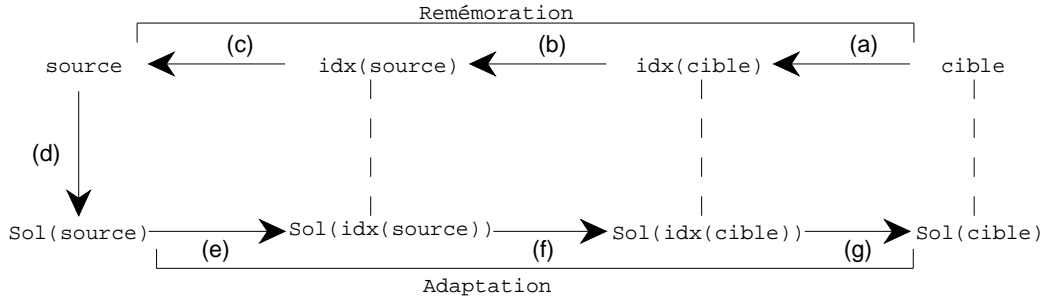


FIG. 3 – Le schéma général de résolution d'un problème cible à partir d'un problème source.

Un cas est un itinéraire noté ($init$, but , $date$, $véhicule$, $segments$, $temps$), où $init$ est la ville de départ, but est la ville d'arrivée, $date$ est la date du voyage, $segments$ est une suite ordonnée de segments intermédiaires, et $temps$ est le temps du trajet. Un *segment* est noté ($départ$, $arrivée$), où $départ$ est la ville de départ et $arrivée$ est la ville d'arrivée. Un problème cible s'exprime sous la forme d'une ville de départ $init_{cible}$ et une ville d'arrivée à atteindre but_{cible} . La solution à un problème cible s'exprime sous forme d'une suite ordonnée de segments $segments$ et d'un temps de trajet $temps$.

La théorie du domaine est décrite par un ensemble de villes A, B, \dots, I , chaque ville V étant associée à l'ensemble de ses villes directement voisines $V' \in voisins(V)$: V' peut être atteinte à partir de V en utilisant un itinéraire élémentaire. L'ensemble des voisins connus est résumé dans la table suivante :

V	Voisins(V)
A	{B, E, G}
B	{A, C, E}
C	{B, D, E}
D	{C, F, I}
E	{A, C, F, G}
F	{E, D, H}
G	{A, E, H}
H	{F, G}
I	{D}

Une distance d entre deux villes V et V' a été définie de la manière suivante :

$$d(V, V') = d(V', V) = \begin{cases} 0 & \text{si } V = V' \\ 1 & \text{si } V \in voisins(V') \\ +\infty & \text{si } V \notin voisins(V') \end{cases}$$

L'étape d'élaboration construit un index du problème cible en ne conservant que les descripteurs $init_{cible}$ et but_{cible} jugés plus pertinents. L'index du cas source est réduit aux seules villes de départ et d'arrivée, les autres descripteurs tels que la date et le véhicules sont jugés moins pertinents.

$$idx(cible) = (init_{cible}, but_{cible})$$

$$idx(source) = (init_{source}, but_{source})$$

De plus, la distance entre un index source et un index cible est définie de la manière suivante :

$$dist(idx(source), idx(cible)) =$$

$$d(init_{source}, init_{cible}) + d(but_{source}, but_{cible})$$

La base de cas contient les cas suivants :

N°cas	idx(source)	sol(source)
1	(A, C)	{((A, B), (B, C)), 2H00}
2	(C, G)	{((C, E), (E, G)), 1H30}
3	(A, D)	{((A, B), (B, C), (C, D)), 3H00}
4	(D, C)	{((D, C)), 1H00}
5	(H, E)	{((H, F), (F, E)), 2H00}
6	(E, B)	{((E, C), (C, B)), 2H15}
7	(G, D)	{((G, E), (E, C), (C, D)), 2H45}
8	(B, I)	{((B, C), (C, D), (D, I)), 3H15}

Soit un problème cible ($B, D, 15/2/99, 2$ CV Citroën). L'élaboration de l'index du problème cible nous donne $idx(cible) = (B, D)$. Cet index est apparié avec les index de cas source de la base de cas précédente, ce qui donne la table des distances suivante :

N°cas	idx(source)	$dist(idx(source), idx(cible))$
1	(A, C)	2
2	(C, G)	$+\infty$
3	(A, D)	1
4	(D, C)	$+\infty$
5	(H, E)	$+\infty$
6	(E, B)	$+\infty$
7	(G, D)	$+\infty$
8	(B, I)	1

Lorsque l'on applique le schéma général de résolution du problème cible de la figure 3, on obtient les étapes suivantes schématisées sur la figure 4 :

- (a) À partir de la description du problème cible ($B, D, 15/2/99, 2$ CV Citroën), l'index du problème cible est construit : $idx(cible) = (B, D)$.

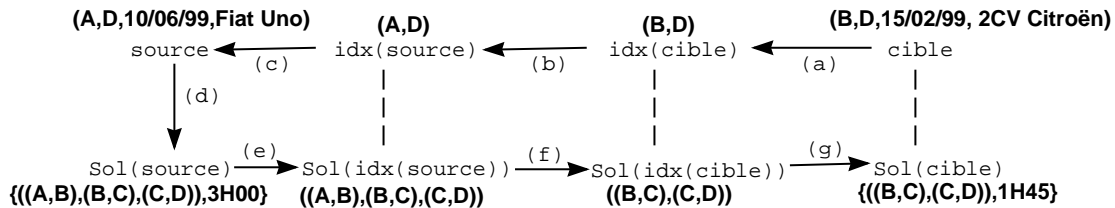


FIG. 4 – Le schéma de résolution du problème en réutilisant le cas numéro 3.

- (b) Un index de cas source est recherché dans la base de cas. Les cas possibles sont le cas 3 (A, D) et le cas 8 (B, I) : tous deux minimisent $\text{dist}(\text{idx}(\text{source}), \text{idx}(\text{cible}))$ de la même façon. $\text{idx}(\text{source}) = (A, D)$ est choisi arbitrairement.
- (c) À partir de l'index (A, D) , le cas $\text{source} = (A, D, 10/06/99, \text{Fiat Uno})$ est accessible.
- (d) La solution $\text{Sol}(\text{source}) = \{((A, B), (B, C), (C, D)), 3H00\}$ de source est réutilisée pour le problème cible.
- (e) La généralisation de $\text{Sol}(\text{source})$ en $\text{Sol}(\text{idx}(\text{source}))$ élimine le descripteur temps : $\text{Sol}(\text{idx}(\text{source})) = ((A, B), (B, C), (C, D))$.
- (f) La solution est spécialisée en $\text{Sol}(\text{idx}(\text{cible})) = ((B, C), (C, D))$ en supprimant le segment (A, B) . Ce type d'opération d'adaptation est décrite en particulier dans [22]. C'est une des opérations de suppression, d'ajout et de substitution qui est licite pour adapter la construction d'une solution.
- (g) La solution $\text{Sol}(\text{idx}(\text{cible}))$ est spécialisée en $\text{Sol}(\text{cible}) = \{((B, C), (C, D)), 1H45\}$, où le descripteur temps a été évalué et rajouté.

7 Discussion et perspectives

Dans cet article, un modèle unifié des étapes de remémoration et d'adaptation en RàPC a été proposé. Ce modèle simple et clair a pu être élaboré grâce à l'expérience accumulée à l'occasion des applications développées dans des domaines variés (voir au paragraphe 3), et même s'il doit encore être approfondi, il remet néanmoins en cause l'idée classique que l'adaptation ne peut être que spécifique à une application particulière. De plus, cette proposition peut servir de base à une discussion de fond concernant deux étapes clé du RàPC. Toutefois, nous avons fait l'hypothèse qu'un cas peut être vu comme un plan de résolution de problème, c'est à dire qu'un état initial, un but et une collection d'opérateurs permettant de générer un espace d'états sont nécessaires. L'adaptation peut alors se résumer à une suite de généralisations,

spécialisations et substitutions effectuées à des niveaux d'abstraction différents au cours du processus de construction d'une solution.

Il faut montrer que l'hypothèse de considérer un cas comme un plan de résolution de problème n'enlève pas de généralité à la notion de cas. Il faut également encore progresser dans la description formelle de ce travail afin de formaliser le processus de remémoration-adaptation indépendamment de tout domaine. Cette description pourrait alors servir de guide pour la conception d'un système de RàPC.

Certains aspects de ce travail doivent encore être détaillés. Tout d'abord, les liens entre les dimensions verticale et horizontale de l'adaptation (paragraphe 4 et 5 respectivement) doivent être plus précisément étudiés. Ensuite, les liens avec les travaux sur l'adaptation de plan doivent être mis en évidence. Enfin, les étapes (a), (b) et (c) de la remémoration et les étapes correspondantes (e), (f) et (g) de l'adaptation doivent être étudiées plus en détail.

Pour compléter cette étude, l'étape d'apprentissage du RàPC doit être intégrée. Dans ce contexte, l'apprentissage n'est pas limité au stockage d'un problème et de sa solution dans la base de cas, mais suppose également l'apprentissage de connaissances de remémoration et d'adaptation à partir de l'analyse des remémorations et des adaptations réalisées, ayant mené à des solutions satisfaisantes ou à des échecs. Différentes directions de recherches peuvent être envisagées, en relation avec les explications qui peuvent être engendrées à partir des succès ou des échecs du processus de résolution de problème (une telle approche est également étudiée dans [13]). Le mécanisme d'apprentissage consiste à faire ressortir une séquence de justifications de succès ou d'échec, qui serviront de règle lors d'adaptations futures et de « modes d'emplois » lors de la réutilisation de cas.

Ce travail de modélisation est issu de deux travaux complémentaires – [27] et [24] – qui reposent sur des applications (DESIGNER, PAD'IM, ACCELERE et SYSEXPRT pour [27], OMEGA et RESYN/RàPC pour [24]). À partir de ce travail, nous envisageons d'étudier d'autres applications du RàPC à la lumière de ce modèle. Une dernière perspective justifiant pleinement ce modèle est son utilisation pour la conception d'applications de RàPC. Ainsi, l'utilité de ce formalisme

et ses limitations pourront être mis en évidence de façon opérationnelle.

8 Conclusion

Dans cet article, des éléments pour une théorie unifiée de l'adaptation en RàPC sont présentés. D'abord, l'hypothèse selon laquelle l'adaptation d'un cas peut se ramener à l'adaptation d'un plan est justifiée. À partir de cette hypothèse, les travaux sur l'adaptation en RàPC peuvent être réutilisés. Puis, l'adaptation d'un cas – vue comme une adaptation de plan – est définie *en parallèle* avec la remémoration de cas. Les étapes d'élaboration du problème cible (ou indexation), de recherche d'index de problème source, et de recherche d'un problème source correspondant à cet index, sont reliées aux trois étapes d'adaptation. Ce point de vue est principalement basé sur l'étude d'applications réelles et possède donc un caractère opérationnel. Il reste encore à préciser et à formaliser cette étude afin de la rendre plus apte à aider à la construction de systèmes de RàPC. De plus, cette étude doit encore prendre en compte l'étape d'apprentissage du RàPC pour compléter le cycle de RàPC.

Références

- [1] R. Bergmann et W. Wilke. PARIS: Flexible Plan Adaptation by Abstraction and Refinement. In *Workshop on Adaptation in Case-Based Reasoning, ECAI-96*, pages 8–12, Budapest, Hungary, August 1996.
- [2] Ralph Bergmann et Wolfgang Wilke. Building and refining abstract planning cases by change of representation language. *Journal of Artificial Intelligence Research*, 3:53–118, 1995.
- [3] L. K. Branting et D. W. Aha. Stratified Case-Based Reasoning: Reusing Hierarchical Problem Solving Episodes. In *14th Joint Conference on Artificial Intelligence, IJCAI-95*, pages 394–390, Montréal, Canada, August 1995.
- [4] J. G. Carbonell. Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In R. Michalski, J. Carbonell, et T. Mitchell, éditeurs, *Machine Learning, An Artificial Intelligence Approach*, volume 2, pages 371–391. Morgan Kaufmann, 1986.
- [5] E. Charniak et D.V. McDermott. *Introduction to Artificial Intelligence*. Addison Wesley, Reading, Massachussets, 1985.
- [6] B. Chiron et A. Mille. Aide à la Conception d'Environnements de Supervision par Réutilisation de l'Expérience. In *Journées Ingénierie des Connaissances et Apprentissage Automatique - JICAA '97*, pages 643–654, Roscoff, France, 20-22 mai 1997.
- [7] B. Fuchs, A. Mille, et B. Chiron. Operator Decision aiding by Adaptation of Supervision Strategies. In M. Veloso et A. Aamodt, éditeurs, *First International Conference on Case-Based Reasoning - ICCBR-95*, pages 23–32, Sesimbra, Portugal, 23–26 octobre 1995. LNAI, vol. 1010, Springer Verlag, Berlin.
- [8] K.J. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, San Diego, 1989.
- [9] S. Hanks et D. S. Weld. A domain independent algorithm for plan adaptation. *Journal of Artificial Intelligence Research*, 2:319–360, 1995.
- [10] O. Herbeaux et A. Mille. ACCELERE: un système d'aide à la conception de caoutchouc cellulaire exploitant la réutilisation de l'expérience. *Journal Européen des Systèmes automatisés*, 1998.
- [11] K. Hua, B. Faltings, et I. Smith. CADRE: Case Based Geometric Design. *Artificial Intelligence in Engineering*, 10:171–183, 1996.
- [12] K. Hua, I. Smith, et B. Faltings. Integrated Case-Based Building Design. In M. M. Richter, S. Wess, K.-D. Althoff, et F. Maurer, éditeurs, *First European Workshop on Case-Based Reasoning - EWCBR-93*, pages 458–469, University of Kaiserslautern, Germany, 1994. LNAI, vol. 837, Springer Verlag, Berlin.
- [13] L.H. Ihrig et S. Kambhampati. Storing and indexing plan derivation through explanation-based retrieval failures. *Journal of Artificial Intelligence Research*, 7:161–198, 1997.
- [14] J. Koehler. Planning from Second Principles. *Artificial Intelligence*, 87:145–186, 1996.
- [15] K. Kumar et B. Raphael. Cadrem: A case-based system for conceptual structural design. *Engineering with computers*, 13(3):153–164, 1997.
- [16] J.E. Laird, A. Newell, et P.S. Rosenbloom. SOAR: An architecture for general intelligence. *AI Magazine*, 33(1):1–64, 1987.
- [17] D. B. Leake, A. Kinley, et D. Wilson. Learning to Improve Case Adaptation by Introspective Reasoning and CBR. In M. Veloso et A. Aamodt, éditeurs, *First International Conference on Case-Based Reasoning - ICCBR-95*, pages 229–240, Sesimbra, Portugal, 1995. LNAI, vol. 1010, Springer Verlag, Berlin.

- [18] D. B. Leake, A. Kinley, et D. Wilson. Acquiring Case Adaptation Knowledge: A Hybrid Approach. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 684–689, Portland, Oregon, 1996. AAAI Press, Menlo Park, CA.
- [19] David Leake, Andrew Kinley, et David Wilson. Case-based similarity assessment: estimating adaptability from experience. In *Proceedings of the 14th National Conference on Artificial Intelligence*, Menlo Park, CA, 1997. AAAI Press, Menlo Park, CA.
- [20] David Leake, Andrew Kinley, et David Wilson. Learning to integrate multiple knowledge sources for case-based reasoning. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1997.
- [21] J. Lieber et A. Napoli. Using Classification in Case-Based Planning. In W. Wahlster, éditeur, *12th European Conference on Artificial Intelligence - ECAI-96*, pages 132–136, Budapest, Hungary, August 1996. J. Wiley and Sons, New York.
- [22] J. Lieber et A. Napoli. Correct and Complete Retrieval for Case-Based Problem Solving. In H. Prade, éditeur, *13th European Conference on Artificial Intelligence - ECAI-98*, pages 68–72, Brighton, United Kingdom, August 1998. J. Wiley and Sons, New York.
- [23] E. Melis. A model of analogy-driven proof-plan construction. In *14th Joint Conference on Artificial Intelligence, IJCAI-95*, pages 182–189, Montréal, Canada, August 1995.
- [24] E. Melis, J. Lieber, et A. Napoli. Reformulation in case-based reasoning. In B. Smyth et P. Cunningham, éditeurs, *Fourth European Workshop on Case-Based Reasoning, EWCBR-98*, LNAI 1488, pages 172–183. Springer Verlag, Berlin, 1998.
- [25] A. Mille, J-L. Di Martino, et A. J. Michel. Adaptation: the key-point in Case Based Reasoning. Case study: Digester Programming Helpin. Workshop on Practical Developments Strategies for industrial strength Case Based Reasoning Applications, 16th International Conference on Artificial Intelligence, IJCAI'95, Montreal, Canada. - Technical report available, 1995. Workshop without proceedings.
- [26] A. Mille, B. Fuchs, et B. Chiron. Raisonement fondé sur l'expérience: un nouveau paradigme en supervision industrielle. *Revue d'intelligence artificielle*, 13:97–128, 1999.
- [27] A. Mille, B. Fuchs, et O. Herbeaux. A Unifying Framework for Adaptation in Case-Based Reasoning. In *Workshop on Adaptation in Case-Based Reasoning, ECAI-96*, pages 22–28, Budapest, Hungary, August 1996.
- [28] A. Newell. Reasoning, problem solving, and decision processes: the problem space as a fundamental category. In R. Nickerson, éditeur, *Attention and Performances VIII*, pages 693–718. Lawrence Erlbaum Associates, Hillsdale, NJ, 1980.
- [29] G. Polya. *How to Solve it*. Double Day Anchor Book, New York, NY, 1957.
- [30] L. Purvis et P. Pu. Adaptation Using Constraint Satisfaction Techniques. In M. Veloso et A. Aamodt, éditeurs, *First International Conference on Case-Based Reasoning - ICCBR-95*, pages 289–300, Sesimbra, Portugal, 1995. LNAI, vol. 1010, Springer Verlag, Berlin.
- [31] I. Smith, C. Lottaz, et B. Faltings. Spatial Composition using Cases: IDIOM. In M. Veloso et A. Aamodt, éditeurs, *First International Conference on Case-Based Reasoning - ICCBR-95*, pages 88–97, Sesimbra, Portugal, 1995. LNAI, vol. 1010, Springer Verlag, Berlin.
- [32] B. Smyth. *Case-Based Design*. Doctoral thesis of the Trinity College, Dublin, 1996.
- [33] B. Smyth et M. T. Keane. Using adaptation knowledge to retrieve and adapt design cases. *Knowledge-Based Systems*, 9(2):127–135, 1996.
- [34] M. Veloso. *Planning and Learning by Analogical Reasoning*. LNAI 886. Springer Verlag, Berlin, 1994.
- [35] A. Voß. How to solve complex problems with cases. *Engineering applications of artificial intelligence*, 9(4):377–384, 1996.
- [36] A. Voß, éditeur. *Proceedings of the ECAI'96 Workshop: Adaptation in Case-Based Reasoning*, 1996.
- [37] A. Voß. Structural Adaptation with topo. In *Workshop on Adaptation in Case-Based Reasoning, ECAI-96*, pages 52–54, Budapest, Hungary, August 1996.
- [38] A. Voß. Case Reusing Systems - Survey, Framework and Guidelines. *Knowledge Engineering Review*, 12(1):59–89, 1997.