

# CIGI 2011

## Modification du cycle du Raisonnement à Partir de Cas pour l'acquisition de connaissances durant l'étape d'adaptation

EDUARDO ROLDAN REYE<sup>1</sup>, STEPHANE NEGNY<sup>1</sup>, JEAN MARC LE LANN<sup>1</sup>

<sup>1</sup> Laboratoire de Génie Chimique UMR 5503-INPT ENSIACET  
4 Allée Emile Monso

31 030 Toulouse Cedex 4, France

eduardo.roldan@ensiacet.fr, stephane.negny@ensiacet.fr, jeanmarc.lelann@ensiacet.fr

---

**Résumé** - Après une introduction pour démontrer la complexité de l'étape d'adaptation en RàPC, cet article détaille une approche basée sur le couplage entre RàPC et programmation par contrainte. Ce couplage est réalisé grâce à une modification du cycle traditionnel de RàPC dans le but d'introduire une boucle d'interaction avec l'expert. Cette interactivité est également exploitée pour acquérir en ligne les connaissances de l'expert nécessaire à la conception de système. En effet, lors d'un échec d'adaptation, l'expert corrige le modèle à l'aide de fonctions d'adaptation qui sont ensuite sauvegardées pour former une méthode d'adaptation. Tout comme le nouveau cas ainsi formé, la nouvelle méthode d'adaptation est stockée dans la mémoire du RàPC. L'objectif principal de l'approche proposée est de réduire l'effort d'extraction, de formalisation, de modélisation et de capitalisation des connaissances. Côté application, cet article traite des problèmes de configuration de systèmes d'agitation mécanique.

**Abstract** - After an introduction to demonstrate the complexity of the adaptation stage in CBR, this article describes an approach based on the coupling between CBR and constraint programming. This coupling is achieved through a modification of the traditional CBR cycle in order to introduce an interaction loop with the expert. This interactivity is also used to purchase online the expert knowledge needed for system design. Indeed, during a failure of adaptation, the expert corrects the model by using adaptation functions which are then stored to form an adaptation method. Like the new case, the new adaptation method is stored in the memory of CBR. The main objective of the proposed approach is to reduce the extraction, formalization, modeling and capitalization of expert knowledge. Concerning application, this article addresses the problems of system configurations for industrial system mixing.

**Mots clés** – Gestion des Connaissances, Conception, RàPC, PPC.

**Keywords** – Knowledge Management. Design, CBR, CSP

---

### 1 INTRODUCTION

L'activité de conception est un processus complexe et décisif. En effet, c'est à travers elle que les solutions à mettre en œuvre sont déterminées afin de réaliser un système (système : produit, procédé, modèle, etc...) conforme aux exigences. De nombreuses décisions sont arrêtées durant cette activité. Remettre en cause l'un des choix génère un allongement de l'activité de conception. Il est donc primordial de réaliser des choix judicieux pour éviter d'itérer et pour converger plus rapidement vers une solution.

Dans le contexte industriel actuel, il est souhaitable de disposer d'approches et d'outils d'aide à la conception (car développer un outil automatique de conception semble un objectif encore utopique) qui permettent de proposer des concepts nouveaux mais aussi de diminuer fortement les délais de l'activité puisque certains choix ne sont plus ni à faire, ni à remettre en cause, notamment lorsqu'il existe une certaine récurrence dans

l'activité. Des approches basées sur le retour d'expériences acquises lors de conceptions antérieures et sur l'exploitation des connaissances et savoirs faire des experts sont d'un grand intérêt.

La mise en place de ces approches et outils passe par différentes étapes telles que la capitalisation des connaissances, leurs formalisations et leurs exploitations. De plus, le cadre de travail doit permettre l'interaction entre le concepteur et l'outil. Le rôle principal du concepteur est de faire des choix, et celui de l'outil est de propager la conséquence de ces choix et de proposer un ensemble d'alternatives.

Parmi l'ensemble des méthodes de gestion des connaissances, le Raisonnement à Partir de Cas (RàPC) et la Programmation Par Contraintes (PPC) sont des méthodes communément utilisées et sans cesse améliorées dans les recherches actuelles. Bien que différentes ces deux approches se révèlent être

complémentaires pour construire un outil d'aide à la conception performant.

Le RàPC ne nécessite pas des connaissances explicites du problème donc nul besoin de les formaliser. En effet, le RàPC repose sur une description précise du problème à l'aide de caractéristiques principales bien identifiées. Cet avantage permet de mettre rapidement en place un outil d'aide à la conception. L'apprentissage par l'introduction de nouveaux cas résolus (ou échecs de résolution) lui confère également un avantage indéniable pour actualiser, expliquer et maintenir la connaissance. En revanche, il faut que la base de cas soit suffisamment fournie pour couvrir l'ensemble des espaces des problèmes et des solutions pour être efficace et obtenir un résultat pertinent. Toutefois, les inconvénients principaux restent la limitation à l'utilisation des cas similaires ce qui ne garantit pas l'optimalité des solutions proposées (discussion, non abordée dans cet article, autour des limites de la mesure de similarité et de l'intérêt d'introduire l'adaptabilité) et son incapacité à établir qu'un problème n'a aucune solution.

Par opposition au RàPC dans les approches basées sur la programmation par contraintes, une analyse préliminaire pour extraire, identifier, modéliser les connaissances et finalement obtenir un modèle valide adapté au problème est indispensable. Cette étape est plus complexe que son équivalente en RàPC et demande également un temps d'analyse et de développement assez long. En contre partie, avec cette approche il est possible d'atteindre des solutions nouvelles voire innovantes mais aussi de démontrer qu'un problème n'a pas de solution. De plus, avec ce type d'approches, l'utilisateur peut interagir avec l'outil. Après chaque décision l'outil lui présente que les choix restant cohérents avec le réseau de contraintes (mode interactif). Cette garantie de cohérence (par respect des contraintes) durant tout le processus de conception réduit ainsi le délai de conception grâce à la limitation des itérations dues à la remise en question de choix précédents (remarque : la cohérence n'impliquant pas une solution satisfaisante). Le RàPC quant à lui retrouvera toujours des cas similaires même si les valeurs des attributs sont incohérentes mais la pertinence de ces cas remémorés n'est pas toujours avérée.

Dans cet article, nous proposons une modification du cycle de RàPC conventionnel afin de coupler les deux approches au niveau de l'étape d'adaptation. Dans la suite, la deuxième partie de cet article revient sur l'étape d'adaptation avec les démarches possibles et le mode d'acquisition des connaissances nécessaires à la réussite de cette étape. La modification du cycle de RàPC avec la modélisation des connaissances de l'expert sous forme de contraintes est détaillée dans la troisième partie. Avant de conclure et de dresser quelques perspectives, la quatrième partie illustre la démarche au travers d'un exemple issu du génie des procédés.

## 2 ADAPTATION EN RÀPC

### 2.1 Le cycle conventionnel du RàPC

Raisonné à partir de cas c'est résoudre le problème cible en s'appuyant sur la base de cas. Actuellement le modèle le plus communément utilisé pour décrire le cycle du RàPC est le modèle  $R^5$  de [Finnie and Sun, 2003], qui est une version étendue (avec l'ajout de la première étape) du modèle  $R^4$  introduit par [Aamodt et Plaza, 1994]. Ce modèle cyclique est illustré sur la figure 1 avec comme étapes :

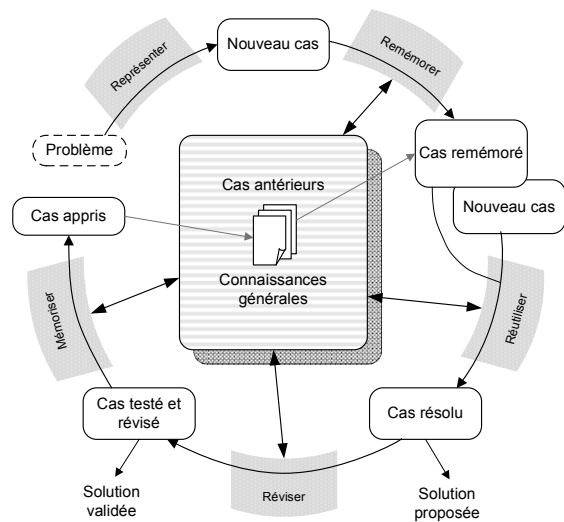


Figure 1 : Cycle du RàPC

- **Représenter** : Il s'agit de mettre en forme le problème cible (problème initial) en vue de la remémoration en complétant sa description à l'aide de la connaissance du domaine. Plusieurs modes de représentation sont envisageables mais la description sous forme d'un vecteur d'attributs valeurs est la plus courante.
- **Remémorer** : Son objectif est de sélectionner et d'extraire un cas similaire de la base (appelé cas source) au problème cible. Le point clef de cette étape est la mesure de similarité. Toutefois le cas le plus similaire n'est pas toujours le plus facile à réutiliser (problème de la mesure de similarité par une distance également nommée similarité de surface). Dans ces conditions il convient d'améliorer cette mesure grâce à des connaissances plus profondes sur les cas en utilisant par exemple une mesure d'adaptabilité (mise en œuvre dans notre système de RàPC mais non abordée dans ce papier).

$(Cible, Base\ de\ Cas) \rightarrow (source, Sol(source)) \in Base\ de\ Cas.$

- **Réutiliser** : En s'appuyant sur le cas remémoré, la réutilisation cherche à résoudre le problème cible.  $(source, Sol(source), cible) \rightarrow Sol(cible)$   
Les approches pour adapter  $Sol(source)$  à cible sont nombreuses. La réutilisation peut être aussi triviale que la proposition directe de  $Sol(source)$  comme solution cible, sans aucun changement (utilisé pour l'aide à la décision ou justifier un choix par exemple). Mais la plupart du temps, cette étape d'adaptation cherche à définir l'écart entre les problèmes source et cible pour ensuite modifier  $Sol(source)$  afin d'obtenir  $Sol(cible)$ . Les méthodes d'adaptation doivent répondre aux deux questions suivantes : Qu'est ce qui devrait être modifié dans  $Sol(source)$  ? Comment ce changement peut il être réalisé ?
- **Réviser** : Suite à l'adaptation,  $Sol(cible)$  proposée est testée, par simulation ou expérimentalement, pour vérifier son adéquation et sa pertinence par rapport à cible. En cas de test négatif, la solution est corrigée pour éliminer les dernières divergences. A noter que ces étapes de test et de réparation peuvent faire partie intégrante de l'adaptation.
- **Retenir** (mémoriser) : S'il est opportun (apport d'une valeur ajoutée à la base de cas), le nouvel épisode est

stocké dans la base. Cette étape fait du RàPC un système auto apprenant, ce qui lui confère l'avantage d'étendre sa couverture de l'espace des problèmes possibles et d'accroître son efficacité. Avec cette étape se pose inévitablement le problème de la maintenance de la base de cas et notamment de la gestion des connaissances d'un tel système.

En réalité, le modèle R<sup>5</sup> est plus complexe que cette présentation rapide car chaque étape englobe de nombreux sous processus spécifiques dernièrement détaillés par [Lopez de Mantaras et al, 2006]

## 2.2 L'adaptation et ses stratégies

L'adaptation est un processus simple à comprendre qui consiste en la transformation de Sol(source) en une solution appropriée Sol(cible). En revanche, l'implantation de ce processus ou simplement sa spécification est plus difficile car elle demande l'ajout de connaissances supplémentaires.

Même si des travaux de recherche visent à réduire l'effort d'adaptation grâce à l'amélioration de l'étape de remémoration, dans de nombreuses applications cette étape demeure cruciale pour assurer l'efficacité d'un système de RàPC. A noter que dans la communauté du RàPC, l'utilité de l'étape d'adaptation a fait débat avec l'opposition de deux visions distinctes. D'un côté certains chercheurs conçoivent des systèmes sans adaptation (ou à faible adaptation), tablant sur l'abondance de cas et leurs capacités à couvrir l'espace des problèmes pour pallier la faiblesse de l'adaptation, de l'autre des chercheurs qui basent leurs études sur cette étape primordiale selon leur point de vue (partagé par les auteurs).

L'adaptation doit répondre à deux questions principales : Que faut il changer dans Sol(source) pour obtenir une solution adéquate ? Et surtout Comment faut il opérer pour procéder à ces modifications ? Pour répondre à cette dernière interrogation, [Lieber et al, 2004] classent les diverses approches complémentaires en trois catégories :

- Les démarches unificatrices. Ces démarches proposent des modèles généraux de l'adaptation sous forme de principes, algorithmes...
- L'acquisition des connaissances d'adaptation. Elles considèrent l'adaptation dans un domaine d'application spécifique et a pour objectif d'extraire, expliciter et modéliser les connaissances dans ce domaine.
- Les catalogues des connaissances. A mi chemin entre les deux approches précédentes, ils formalisent et utilisent des connaissances d'adaptation applicables à plusieurs domaines [Lieber, 2008].

Dans cet article nous proposons une méthode qui permet de réduire ce travail d'acquisition et de modélisation des connaissances d'adaptation. Mais dans un premier temps, il faut mettre en évidence les stratégies sur l'adaptation.

Concernant l'adaptation d'un cas, il est possible de distinguer trois stratégies majeures :

- **Ré-instanciation** : c'est la forme la plus simple d'adaptation, dans laquelle Sol(source) est simplement retournée et utilisée directement sans modification.
- **Substitution** : elle remplace les valeurs des attributs de Sol(source) qui sont inacceptables car en conflit ou contradictoires avec les exigences de Pb(cible).
- **Transformation** : elle est employée quand aucune substitution appropriée n'est possible. Une solution

sera dérivée en se basant sur les contraintes et les caractéristiques de la solution voulue. Elles spécifient quelles propriétés la solution devrait ou ne devrait pas avoir.

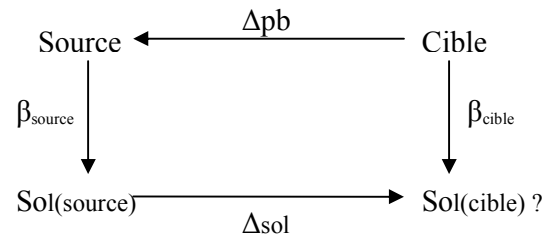
Il est à noter que ces stratégies sont complémentaires. En effet en fonction du degré de similitude entre le cas source et le cas cible, on optera pour l'une ou l'autre des stratégies. De part l'effort demandé, la transformation est sûrement la plus complexe et délicate car dans cette situation, la solution source devra être ajustée, en partie ou totalement.

## 2.3 Les démarches

Le raisonnement analogique a un rôle central dans le raisonnement, c'est un processus d'identification et de mobilisation des connaissances qui ont été extraites d'une situation familière, afin de résoudre une situation méconnue. Dans ces conditions, l'adaptation est considérée comme un raisonnement par analogie intra-domaine par opposition à un raisonnement par analogie inter domaine. Ceci implique que pour un système de RàPC, il existe un domaine d'application bien déterminé auquel appartiennent tous les cas. Le problème d'adaptation a été symbolisé par [Chouraqui, 1986] par le diagramme de la figure 2, communément appelé « carré d'analogie ». Sur cette figure les  $\Delta$  sont les liens entre les cas source et cible. Ils matérialisent entre autres, les similarités et les différences entre ces deux cas.  $\Delta_{pb}$  contient les informations sur les liens entre les problèmes source et cible (souvent nommé appariement entre problèmes). Une relation de dépendance  $\beta$  traduit le lien entre un problème et sa solution, source intervient dans la déduction de Sol(source).

A partir de ce « carré d'analogie », deux lectures sont possibles pour effectuer l'adaptation :

- Adaptation par transformation : elle s'appuie sur l'analogie transformationnelle qui essaie de réutiliser avec quelques modifications, une solution précédemment trouvée dans une situation similaire. Elle cherche à construire  $\Delta_{sol}$  à partir de  $\Delta_{pb}$  (et éventuellement Sol(source)) et de connaissances supplémentaires.
- Adaptation par dérivation : elle adapte le processus de raisonnement qui a amené à la résolution du problème similaire. Chaque étape du raisonnement menant à la solution est mémorisée et stockée (avec les informations nécessaires pour sa compréhension : décisions prises, connaissances utilisées...). Dans cette méthode, c'est  $\beta_{cible}$  qui est recherchée et élaborée à partir des informations connues sur  $\beta_{source}$ .  $\Delta\beta$  représente les modifications à apporter à  $\beta_{source}$  pour obtenir  $\beta_{cible}$ .



**Figure 2 : Carré d'analogie**

[Lieber, 2008] résume ces deux modes d'adaptation à l'aide des étapes suivantes :

**Transformation :**  
 AT-1 (Source, Cible) →  $\Delta_{pb}$   
 AT-2 (CA,  $\Delta_{pb}$ ) →  $\Delta_{sol}$

AT-3 (Sol(source),  $\Delta\text{sol}$ )  $\rightarrow$  Sol(cible)

**Dérivation :**

AD-1 (Source, Sol(source))  $\rightarrow \beta_{\text{source}}$

AD-2 (Source, Cible)  $\rightarrow \Delta\text{pb}$

AD-3 (CA,  $\Delta\text{pb}$ )  $\rightarrow \Delta\beta$

AD-4 ( $\beta_{\text{source}}$ ,  $\Delta\beta$ )  $\rightarrow \beta_{\text{cible}}$

AD-5 (Cible,  $\beta_{\text{cible}}$ )  $\rightarrow$  Sol(cible)

CA : Connaissances d'adaptation

A noter que les étapes de AD-2 jusqu'à AD-4 correspondent à de l'adaptation par transformation sur les dépendances et non plus sur la solution.

Lorsqu'elle devient complexe, l'adaptation est décomposée en étapes successives. Deux approches sont alors possibles : dans la décomposition horizontale source est relié à cible via un chemin de similarité et Sol(source) à Sol(cible) par un chemin de modification (décomposition le long des  $\Delta$ ). La décomposition verticale, quant à elle, dissocie l'action conjuguée des différents appariements sur les problèmes et les dépendances.

#### 2.4 Connaissances d'adaptation

L'acquisition des connaissances d'adaptation (notée ACA par une partie des chercheurs de la communauté française) tend à extraire des connaissances d'adaptation spécifiques à un domaine donné et à les modéliser. [Lieber et al, 2004] dressent une comparaison entre les différentes méthodes pour cette acquisition. Avant de se poser la question sur la représentation de ces connaissances, il est utile de regarder les sources utilisées pour les extraire.

Selon ce critère, deux approches principales se détachent pour acquérir ces connaissances d'adaptation. L'approche traditionnelle consiste à coder manuellement cette connaissance issue d'interview d'experts dans le domaine d'application (nommée ACA supervisée). Cette connaissance est souvent représentée sous forme d'arbres de décision, de règles...

Dans une approche alternative, la connaissance est directement extraite des bases de connaissances, essentiellement la base de cas, par des techniques de « machines apprenantes » (machine learning) ou des techniques d'extraction de connaissances dans des bases de données. Ces techniques d'apprentissage automatique reposent sur des cas d'adaptation déjà disponibles dans la base ou générés à partir de la base de cas. Pour la deuxième possibilité, deux cas sources sont extraits de la base (Source<sub>1</sub>, Sol(source<sub>1</sub>)) et (Source<sub>2</sub>, Sol(source<sub>2</sub>)), le principe consiste à identifier la connaissance d'adaptation recherchée en supposant que Sol(Source<sub>2</sub>) peut être atteinte à partir de (Source<sub>1</sub>, Sol(source<sub>1</sub>)) et Source<sub>2</sub>. Par exemple en se basant sur le mode d'adaptation par transformation, cette approche est une autre exploitation de l'étape AT-2. En temps normal (réalisation de l'adaptation), CA et  $\Delta\text{pb}$  sont connus et  $\Delta\text{sol}$  est déterminé. Lorsqu'il s'agit d'acquérir les connaissances d'adaptation alors,  $\Delta\text{pb}$  et  $\Delta\text{sol}$  sont connus et CA est à déterminer. Ensuite parmi l'ensemble des couples (Source<sub>1</sub>, Sol(source<sub>1</sub>)), (Source<sub>2</sub>, Sol(source<sub>2</sub>)) retenus (tous ou seulement un certain nombre sur la base d'une mesure de dissimilarité), il faut réaliser des regroupements et généralisations pour arriver aux connaissances d'adaptation et finalement les soumettre à un expert du domaine pour validation.

Acquérir la connaissance d'adaptation à partir de l'expert métier nécessite un travail considérable d'extraction et de

modélisation des connaissances qui peut se révéler très complexe et coûteux en temps. De plus la question de la maintenance de la connaissance acquise se pose inéluctablement. Par contre ces connaissances ont l'avantage d'être facilement compréhensibles car elles sont souvent accompagnées d'explications. A l'inverse, les connaissances d'adaptation générées automatiquement ont l'avantage d'être facilement produites (algorithme mathématique de génération automatique) mais leurs compréhensions restent souvent difficiles. Elles sont d'autant moins exploitées qu'elles ne sont jamais accompagnées d'explication ce qui complique la justification et la pertinence des modifications et des décisions prises (ce qui peut s'avérer gênant dans certains domaines d'application comme la conception par exemple avec les conséquences que peuvent avoir de mauvaises décisions). Que se soit automatiquement ou grâce à l'intervention d'un expert, une fois les connaissances d'adaptation acquises, il faut les formaliser. Bien que plusieurs formalismes soient décrits dans la littérature, les règles et les cas d'adaptation représentent les deux formes les plus couramment implémentées dans les systèmes de RàPC (les règles étant très souvent appliquées pour acquérir les connaissances auprès des experts). Dans la suite nous proposons de modéliser les connaissances de l'expert sous forme de contraintes.

#### 2.5 Couplage RàPC-PPC

Après avoir brièvement comparés et opposés le RàPC et l'approches par PPC, il paraît intéressant de les coupler de par leurs qualités complémentaires en vue de construire un outil d'aide à la conception. Bien que ce soient des méthodes de résolution de problème, elles reposent sur des connaissances différentes : implicites (mais explicites également possible) pour le RàPC, explicites pour la programmation par contraintes. [Codet de Boisse et al, 2010] ont identifié sept types de couplage possibles entre RàPC et PPC : 2 visant à valider les connaissances stockées soit dans l'un soit dans l'autre, 2 visant à modifier ou créer de la connaissance soit dans la PPC soit dans le RàPC, 2 utilisant les approches séquentiellement pour aider à la décision et enfin un dernier type utilisant les approches pour combiner les connaissances générales et contextuelles (objet de leur étude).

Dans le cadre de notre étude nous nous plaçons dans une approche séquentielle avec le RàPC pour extraire le (ou les) cas le plus similaire et la PPC pour aider la phase d'adaptation. Certains auteurs comme [Purvis, 1998], [Pralus et Geneste, 2007] ont déjà intégré la programmation par contraintes au niveau de l'étape d'adaptation. La programmation par contraintes est intégrée comme une méthode d'adaptation qui nécessite donc la formalisation et la modélisation des connaissances (on retrouve les écueils des méthodes d'adaptation présentées précédemment, i.e. travail considérable d'extraction et de formulation).

Cependant, les inconvénients majeurs des approches précédentes comme par exemple le manque d'interactivité et de cohérence pour le RàPC d'une part, et le gros effort d'abstraction pour la programmation par contraintes d'autres part ne sont pas résolus et limitent l'efficacité du couplage proposé. Cet article propose un couplage plus fort entre ces deux approches avec la mise en place d'une boucle d'interaction supplémentaire avec l'expert dans le système de RàPC.

### 3 INTERACTION AVEC L'EXPERT

#### 3.1 Acquisition des connaissances

Dans le système de RàPC proposé, la partie solution d'un cas comprend non seulement la (ou les) solution(s) au problème de conception mais aussi l'ensemble du raisonnement ayant conduit à celle-ci modélisé sous la forme d'un PPC. Le système inclut également une boucle d'interaction avec l'expert au niveau de l'étape d'adaptation et plus particulièrement lors de la révision du cas. En effet, les méthodes d'adaptation présentées dans le paragraphe précédent se déroulent toutes hors ligne, c'est-à-dire hors épisode de résolution. L'acquisition en ligne (ou également appelée opportuniste) des connaissances consiste à profiter de la révision pour solliciter ponctuellement l'expert afin d'acquérir ses connaissances. Pour ce faire, le cycle traditionnel du RàPC a été modifié en s'appuyant sur le cycle « re revisité » proposé par [Cordier et al, 2007].

Les premières étapes du cycle sont identiques à celles du RàPC traditionnel. La phase de remémoration du cas procède au choix d'un cas source. A ce cas source est associé une ou plusieurs méthodes d'adaptation qui décompose l'ensemble des modifications à apporter au PPC du problème source pour la résolution éventuelle du problème cible. Suite à la présentation des différentes méthodes d'adaptation stockées avec le cas source, l'utilisateur a donc la possibilité d'activer une méthode pour la conserver telle que ou la modifier et ainsi construire son propre modèle de résolution :

Adaptation : ( $\beta_{source}$ , Méthode d'adaptation choisie)  $\rightarrow$   $\beta_{cible}$  (correspond à l'étape AD4 dans laquelle  $\Delta\beta$ =Méthode d'adaptation).

Les connaissances de l'expert sont nécessaires à la construction de la méthode d'adaptation.

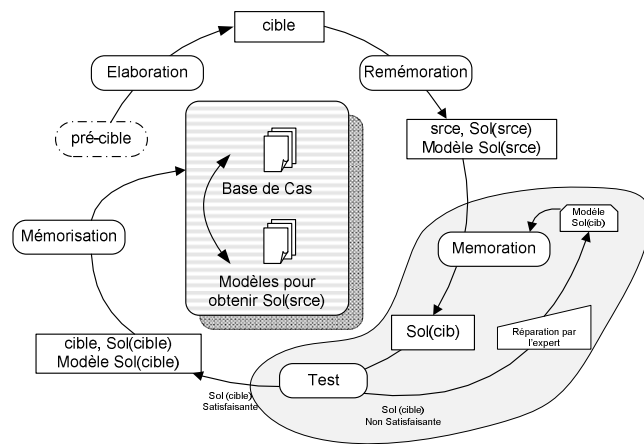


Figure 3 : Boucle d'interaction avec l'expert

Après la construction de  $\beta_{cible}$ , la solution obtenue pour Sol(cible) par résolution du PPC est alors proposée à l'expert lors de l'étape de test. Suite à la phase de tests deux options sont alors envisageables : soit Sol(cible) convient et le cycle continue normalement avec le stockage du cas cible et de la méthode d'adaptation, soit Sol(cible) ne convient pas et la boucle d'interaction avec l'expert est activée. Le modèle utilisé pour résoudre cible est modifié par l'expert afin qu'il corresponde à ses attentes. Ces modifications sont apportées par l'intermédiaire de fonctions d'adaptation permettant d'ajouter, modifier, supprimer des variables, des contraintes, des domaines de définition, des valeurs dans ces domaines. Cette approche présuppose que l'adaptation puisse être

décomposée en un ensemble fini d'opérations successives de modification.

A la fin de chaque modification, la nouvelle PPC ainsi créée est résolue puis testée afin de vérifier si la solution obtenue est acceptable. Cette boucle de test/réparation reste active tant qu'une solution validée n'ait été trouvée. Ces interactions avec l'expert (via les fonctions) sont l'occasion d'acquérir des nouvelles connaissances sur le système à concevoir et contribuent ainsi à diminuer l'effort d'ingénierie des connaissances (très pénalisant dans une approche de PPC seule).

A noter que les leviers d'action de l'expert ne portent que sur la modification des éléments du modèle. A l'heure actuelle, il ne peut pas remettre en cause la décomposition hiérarchique proposée avec  $\beta_{source}$ . Cet aspect sur la proposition des fonctions de plus haut niveau offrant la possibilité de modifier la hiérarchie et d'en tenir compte dans la méthode d'adaptation est une perspective de ce travail.

#### 3.2 Boucle d'interaction

Lorsque la boucle est activée c'est que la solution est jugée non satisfaisante. Il faut donc identifier les raisons de l'échec. Les sources d'échec sont multiples : modèle incohérent car sur contraint, solutions admissibles trop nombreuses, modèle non adapté au problème initial... Dans ce dernier cas, l'échec peut provenir d'une ou plusieurs fonctions d'adaptation voire d'une ou plusieurs contraintes. La correction à apporter au modèle peut se réaliser de deux façons complémentaires :

- Grâce à ses connaissances l'expert identifie les corrections à apporter. Des commentaires explicatifs obligatoires pour expliquer le choix de chaque élément des parties générique et spécifique du modèle mais aussi de chaque fonction de modification sont une aide pour son analyse. Ces commentaires ont pour vocation principale d'expliquer la connaissance acquise en vue de sa future réutilisation.
- Si après le point précédent l'utilisateur est toujours dans l'impasse, le système lui propose de partir du modèle générique puis d'ajouter progressivement les éléments de la partie spécifique puis les modifications, figure 4. Ces approches par étapes successives calquée sur celle de [Cordier et al, 2007] (à la différence majeure que dans notre approche les éléments sont inclus graduellement et testés avec l'ensemble du modèle, et non pas de façon isolée) conduit à l'identification de la (ou des) source d'échec. Après l'ajout d'un nouvel élément, le modèle est résolu puis validé par l'expert (validation intermédiaire). Si un problème est détecté, l'expert peut ainsi le corriger. Suite à la validation intermédiaire, il est possible soit d'ajouter un nouvel élément pour compléter le modèle puis de recommencer la suite de la boucle, soit de valider le modèle lorsque celui fournit une solution adéquate.

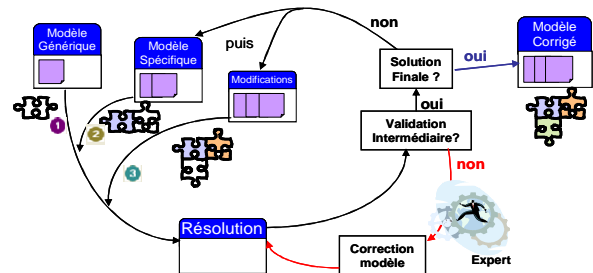


Figure 4 : boucle d'aide à la correction du modèle

### 3.3 L'outil

L'outil construit sur cette approche permet d'interagir avec le concepteur, il lui offre également la possibilité de représenter et manipuler son problème. Les différentes parties permettant ces possibilités sont séparées les unes des autres, figure 5. Il propose donc un cadre de travail dont la structure repose sur les principes suivants :

- La modélisation et la résolution sont disjointes.
- De même au sein du module de modélisation, les contraintes portant sur les connaissances génériques et celles sur les connaissances spécifiques à la définition du problème (exigences ou point de vue du concepteur) sont clairement dissociées. Lors de l'affichage de la méthode d'adaptation, les deux types de contraintes sont présentées avec l'opportunité offerte à l'utilisateur d'inclure ou non les contraintes spécifiques (car souvent dédiées à un problème donné).
- La décomposition hiérarchique d'un système se réalise dans un module dédié, et est applicable à tout type de systèmes. Par contre comme précisé précédemment, elle est figée avec la méthode d'adaptation. L'utilisateur ne peut pas modifier cette structure, si nécessaire il doit bâtir sa propre méthode d'adaptation à partir de zéro pour pouvoir revoir la décomposition. La possibilité d'inclure la modification de la décomposition dans la méthode d'adaptation (modification plus profonde du modèle) reste une perspective d'amélioration.

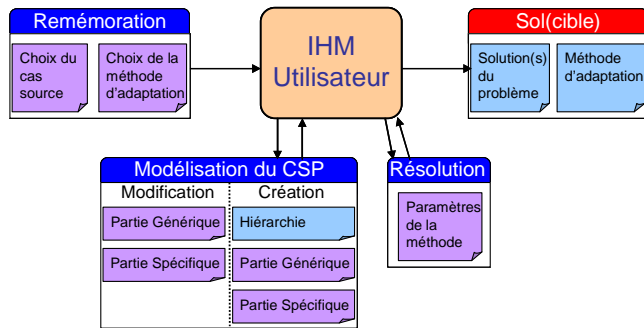


Figure 5 : Schéma de mise en œuvre de la structure

## 4 APPLICATION

### 4.1 Mélangeur

En génie des procédés, l'agitation est une opération importante pour créer des conditions favorables pour les transferts de quantité de mouvement, de chaleur et de matière. Elle est requise à de très nombreux stades des procédés : du stockage des matières premières et produits finis, en passant par des opérations de réaction, de mise en contact de phases... L'objectif d'un mélangeur est variable et plus ou moins facilement atteignable : homogénéisation, transfert thermique et/ou de matière, mise en suspension, dispersion, émulsification... Un mauvais choix d'agitateur, des mauvais dimensionnements peuvent conduire à des erreurs de conception sur cette opération se traduisant par des mauvaises caractéristiques sur les courants de sortie avec des conséquences parfois désastreuses sur la suite du procédé.

Parmi l'ensemble des possibilités d'agitation pouvant assurer le mélange, ce travail se limite aux systèmes mécaniques par

rotation (à l'heure actuelle, la plupart des opérations d'agitation s'effectuent avec ce type de technologies). En première approche, le dispositif d'agitation mécanique est supposé se composer de trois parties : l'entraînement, le (ou les) mobile d'agitation et enfin la cuve. Les choix technologiques possibles pour effectuer un mélange atteignant les objectifs voulus sont nombreux et dépendent de la nature et des propriétés des phases présentes (type de fluides : Newtonien ou non, nature des phases : liquide, liquide-liquide miscibles ou non, gaz-liquide...), de l'application visée qui va réclamer des caractéristiques d'écoulements spécifiques au sein de la cuve. En considérant tous ces éléments, le concepteur doit choisir parmi une grande variété d'agitateurs classés en fonction du flux liquide généré (écoulement radial, axial ou tangentiel). Il doit également fixer la géométrie de la cuve et la présence ou non de systèmes annexes (échange thermique, chicane...). Tous ces choix demandent une certaine expertise.

### 4.2 Modèle pour la configuration d'un système d'agitation

Afin de répondre aux exigences de conception, le modèle générique nécessite des données d'entrée provenant soit des besoins du concepteur, soit de calculs préliminaires (à noter que dans cette phase de conception préliminaire ces données sont souvent méconnues précisément mais seul un ordre de grandeur voire une information qualitative sont disponibles). Dans le cas d'un mélangeur, les données à fournir par l'utilisateur concernent les phases à mélanger, le type d'opération, les caractéristiques hydrodynamiques recherchées et la présence éventuelle d'un système thermique, figure 6.

Phases		
Etat	[Menu déroulant]	
Données	[Menu déroulant]	
Liquide	Masse Volumique	[Champ] kg/m <sup>3</sup>
	Viscosité	[Menu] ou [Champ] Pa.s
	Type Fluide	[Menu]
Solide	Masse Volumique	[Champ] kg/m <sup>3</sup>
	Mouillabilité	[Menu]
	Solubilité	[Menu]
	Diamètre Moyen	[Menu] ou [Champ] mm
Gaz	Débit	[Menu] ou [Champ] m <sup>3</sup> /s
	Pression	[Menu] ou [Champ] Pa
	Solubilité	[Menu]
	Ciel Gazeux	[Menu]
Type Opération		
Type Application	Caractéristique Physique	Caractéristique Chimique
Liquide-Solide	Suspension	Dissolution/Christallisation, Extraction liquide solide, lixiviation, polymérisation, réaction avec catalyseur, stockages de pulpes ou pâte...
Liquide- Gaz	Dispersion	Fermentation, Absorption, Aération, Oxydation, Ozonation, Désorption, Réaction...
Liquide-Liquide	[Menu]	Homogénéisation, Dilution, Neutralisation, Réaction Chimique, Extraction Liquide-Liquide...
Liquide/Solide	Melange (stockage)	Transfert Thermique, Circulation...
Caractéristique Hydrodynamique		
	Cisaillement	[Menu]
	Turbulence	[Menu]
	Pompage	[Menu]
Thermique		
	Echange	[Menu]

Figure 6 : Données du Problème de PPC

Il faut ensuite préciser les variables et de leurs domaines de définition. Dans le cas présent, l'architecture du système technique « Mélangeur » est composé de trois branches principales Cuve, Mobile et Entraînement. Chacune de ces branches englobe un certain nombre de variables accompagnées de leurs domaines associés, figure 7.

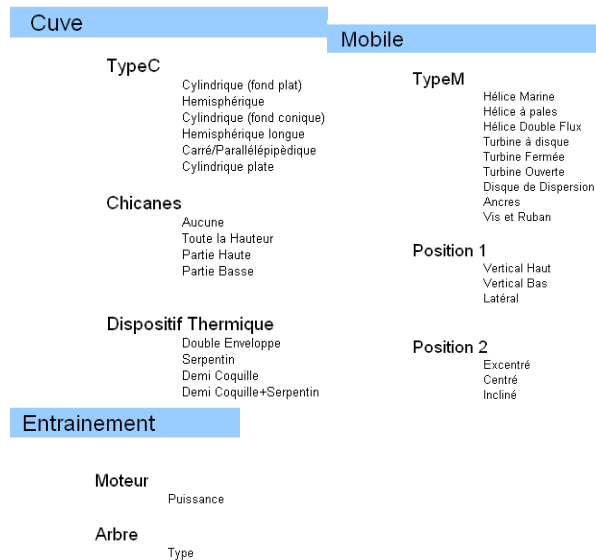


Figure 7 : Variables et Domaines du Modèle Générique

Pour compléter le modèle, les contraintes sont ajoutées, elles sont parfois enrichies d'un commentaire pouvant expliquer l'intérêt de cette contrainte (une connaissance commentée est souvent plus crédible et réutilisable). La figure 8 ci-dessous présente une extraction de quelques contraintes du modèle complet.

C(Etat; Mouillabilité; Diamètre; Chicane)				
Etat	Mouillabilité	Diamètre	Chicane	Commentaire
Liquide-solide	Faible	Faible	Aucune	Présence d'un vortex qui a tendance à aspirer vers l'intérieur les particules de solide qui resteraient à la surface
Liquide-solide	Moyen	Moyen	Partie Haute	Pas de chicane en bas pour faciliter la mise en suspension des particules en fond de cuve
Liquide-solide	Fort	Moyen	Partie Haute	
Liquide-solide	Moyen	Fort	Partie Haute	
Liquide-solide	Fort	Fort	Partie Haute	

C(Etat; TO; Cisail; Turb; Pomp; viscositéL; TypeM)							
Etat	TO	Cisail	Turb	Pomp	viscosité L	Type M	
Liquide-Solide	Suspension	Faible	Faible	Fort	Faible	Hélice Marine	
Liquide-Solide	Suspension	Faible	Faible	Fort	Faible	Turbine Ouverte	
Liquide-Solide	Suspension	Faible	Faible	Fort	Faible	Hélice à pales	
Liquide-Solide	Suspension	Faible	Faible	Fort	Faible	Turbine fermée	
Liquide-Solide	Suspension	Moyen	Moyen	Fort	Faible	Hélice Marine	
Liquide-Solide	Suspension	Moyen	Moyen	Fort	Faible	Turbine Ouverte	
Liquide-Solide	Suspension	Moyen	Moyen	Fort	Faible	Hélice Double Flux	
Liquide-Solide	Suspension	Moyen	Moyen	Fort	Faible	Hélice à pales	
Liquide-Solide	Suspension	Moyen	Moyen	Fort	Faible	Turbine fermée	
Liquide-Solide	Suspension	Fort	Moyen	Fort	Faible	Disque de Dispersion	
Liquide-Solide	Suspension	Fort	Faible	Moyen	Faible	Turbine fermée	
Liquide-Liquide Miscible	Melange	Faible	Faible	Fort	Faible	Hélice Marine	
Liquide-Liquide Miscible	Melange	Faible	Faible	Fort	Faible	Turbine Ouverte	
Liquide-Liquide Miscible	Melange	Faible	Faible	Fort	Moyen	Hélice Double Flux	
Liquide-Liquide Miscible	Melange	Faible	Faible	Fort	Faible	Hélice à pales	
Liquide-Liquide Miscible	Melange	Faible	Faible	Fort	Moyen	Vis et Ruban	
Liquide-Liquide Miscible	Melange	Faible	Faible	Fort	Faible	Ancres	
Liquide-Liquide Immiscible	Dispersion	Fort	Fort	Moyen	Faible	Hélice à pales	
Liquide-Liquide Immiscible	Dispersion	Fort	Fort	Moyen	Faible	Turbine à disque	
Liquide-Liquide Immiscible	Dispersion	Fort	Fort	Moyen	Faible	Turbine fermée	
Liquide-Liquide Immiscible	Dispersion	Fort	Fort	Moyen	Faible	Disque de Dispersion	
Liquide-Liquide Immiscible	Dispersion	Fort	Fort	Moyen	Faible	Hélice à pales	
Liquide-Liquide Immiscible	Emulsion	Fort	Fort	Moyen	Faible	Turbine à disque	
Liquide-Liquide Immiscible	Emulsion	Fort	Fort	Moyen	Faible	Turbine fermée	
Liquide-Liquide Immiscible	Emulsion	Fort	Fort	Moyen	Faible	Disque de Dispersion	
Liquide-Gaz	Dispersion	Fort	Fort	Fort	Faible	Turbine Ouverte	
Liquide-Gaz	Dispersion	Fort	Fort	Fort	Moyen	Hélice Double Flux	
Liquide-Gaz	Dispersion	Fort	Fort	Fort	Faible	Hélice à pales	
Liquide-Gaz	Dispersion	Fort	Fort	Fort	Faible	Turbine à disque	

- C(Etat ; Ciel; Solubilité; Chicane)
- C(TypeC; Chicane)
- C(ViscositéL; Chicane)
- C(Echange; Dispo Th)
- C(Etat; TO; TypeC)
- C(TypeC; Chicane ; Position1; Position 2)

Autres contraintes :

Figure 8 : Exemples de contraintes du modèle Générique

Suite à la remémoration, le cas source sélectionné fournit le modèle générique présenté dans la partie précédente. Ce cas contient également trois méthodes d'adaptation différentes. Deux d'entre elles sont rapidement éliminées par l'expert grâce à une mesure de pertinence sur ces méthodes d'adaptation. En effet, le système de RàPC comporte également une mesure qui permet de quantifier la pertinence de chaque méthode en fonction des objectifs de conception et du problème initial. Ainsi dans l'exemple proposé les deux méthodes éliminées comportent des contraintes trop spécifiques (contraintes pour imposer le type de cuve dans un cas, contrainte sur des conditions d'écoulement très particulières pour le second). La méthode d'adaptation retenue est composée de 7 fonctions d'adaptation successives :

- 1- Ajouter une valeur dans le domaine TypeM (type mobile)
- 2- Ajouter un type d'arbre d'agitation (Entraînement)
- 3- Ajouter une contrainte sur la position 1 : Privilégier la position verticale haute car la plus utilisée

4- Imposer une cuve Cylindrique à fond plat

Pour cette contrainte, le concepteur avait comme objectif de réutiliser un appareillage déjà présent dans son atelier.

5- Ajouter une contrainte sur la puissance moteur

Contrainte souvent nécessaire car il faut souvent trouver un juste équilibre entre les caractéristiques du mélange et le coût énergétique de l'opération dû à l'entraînement de l'agitateur.

6- Ajouter une contrainte de non sélection des mobiles d'agitation du type vis et ruban

Souvent très spécifique à des types de fluides particuliers

7- Supprimer les valeurs « Excentré » et « Incliné » dans le domaine de position 2 :

Même si elles procurent des conditions hydrodynamiques particulières dans la cuve ces deux positions de l'agitateur ont pour conséquence d'augmenter fortement la puissance du moteur et d'accroître les contraintes mécaniques sur l'arbre.

A la lecture de cette méthode et avant résolution du modèle complet de PPC (modèle générique et modèle spécifique constitué des changements apportés par la méthode d'adaptation), l'expert décide de garder les points 1 et 2 car ils élargissent le domaine de définition de certaines variables (non exhaustivité lors de la définition du problème initial). Le point 3 est gardé également. Quant à la fonction 4, elle est supprimée car très spécifique au problème précédemment résolu avec cette méthode et non valable dans le problème présent (surtout que dans la majorité des cas les cuves sont hémisphériques pour des raisons d'écoulement et de vidange). La contrainte du point 5 est conservée par contre la valeur de la puissance maximale a été revue à la hausse pour être adaptée au cas traité. Les contraintes 6 et 7 sont également maintenues, surtout la dernière qui est en cohérence avec celle du point 5. Toujours avant exécution du modèle une contrainte supplémentaire est ajoutée concernant le dispositif thermique. En effet le problème traité nécessite une forte évacuation de la chaleur au niveau de la paroi, dans ces conditions il vaut mieux privilégier des dispositifs de type double enveloppe ou demi-coquille.

Le modèle ainsi construit pour traiter le problème courant est ensuite résolu fournissant un nombre trop important de configurations possibles, dont 4 sont complètement inadaptées au cas présent. Dans ces conditions la boucle avec l'expert est activée et le problème est progressivement amélioré, en

ajoutant des contraintes supplémentaires dans un premier temps et en modifiant la contrainte sur la puissance moteur dans un second temps (diminution de la valeur trop optimiste au départ). Au final le modèle donne le choix entre 3 configurations envisageables qu'il faudra étudier en conception détaillée.

La nouvelle méthode ainsi construite est sauvegardée dans la base en la liant avec le cas source mémorisé. Ce cas source contiendra alors 4 méthodes d'adaptation dont deux d'entre elles (celle réutilisée et la nouvelle) contiennent l'ajout de valeurs dans les domaines TypeM, et Arbre. Ces deux nouvelles valeurs sont alors incluses directement dans le modèle générique et par conséquent supprimées dans les méthodes précédentes. Cet ajustement du modèle générique en fonction des méthodes d'adaptation pose la question de la maintenance des connaissances ainsi acquises. Cette discussion sera abordée dans la conclusion.

## 5 CONCLUSION

Dans ce chapitre, il a été développé une contribution à l'amélioration de l'acquisition des connaissances de conception auprès d'un expert. L'approche proposée est née de la volonté de réduire l'effort d'ingénierie des connaissances. Elle repose sur le couplage entre les méthodes de RàPC et de programmation par contraintes. Couplage réalisé grâce à une modification du cycle traditionnel du RàPC pour mettre en place une boucle d'interaction avec l'expert afin d'acquérir ses connaissances durant un épisode de résolution. Cette approche a été testée sur une application en génie des procédés : conception d'un mélangeur.

Bien que l'effort d'extraction, formalisation et modélisation des connaissances soit réduit par rapport à une approche par programmation par contraintes seule, il faut toutefois une analyse minimale préalable afin d'alimenter la base de cas avec des cas sources et par conséquent avec des modèles de conception basés sur la PPC. En adjoignant des méthodes d'adaptation aux cas sources, se pose inévitablement la question de la gestion et de la maintenance de ces méthodes et des connaissances : Lorsqu'une nouvelle méthode est créée suite à une nouvelle résolution faut-il l'ajouter au cas source mémorisé ou faut-il créer un nouveau cas ? Comment gérer les différentes méthodes d'adaptation associées à un cas source ? Comment gérer les connaissances comprises dans les méthodes d'adaptation ? Les réponses à ces questions font partie des perspectives à donner à ce travail.

En termes d'application, cette approche est une première brique pour la construction d'un outil d'aide à la conception dédié à l'intensification des procédés. Une grande variété d'opérations intensifiées développées aussi bien dans le monde académique que dans le monde industrielle crée un large panel d'options pour potentiellement améliorer les procédés. L'idée est donc de présenter un cadre de travail pour la synthèse et la conception des possibilités d'intensification grâce à une étape d'analyse du problème, la génération des options envisageables, la proposition d'une pré-configuration de l'appareillage et finalement une étape d'évaluation des différents concepts retenus.

## 6 RÉFÉRENCES

Aamodt A., Plaza E., (1994) Case-based reasoning: foundation issues, methodological variations and system approaches, *Artif. Intel. Comm.* 7, pp 39-59.

Chouraqui E., (1986) Le raisonnement analogique : sa problématique, ses applications. Actes des Journées Nationales sur l'Intelligence Artificielle, Aix-les-Bains, CEPADUES-Editions, Toulouse, pp 107-117.

Codet de Boisse A., Vareilles E., Aldanondo M., Gaborit P., Coudert T., Geneste L., (2010) Couplage csp et cbr : premières identifications des modes de couplage. *MOSIM 2010*, pp 759-767.

Cordier A., Fuchs B., Lieber J., Mille A., (2007) Acquisition interactive des connaissances d'adaptation intégrée aux sessions de raisonnement à partir de cas—Principes, architecture IAKA et prototype KAYAK, 15ème Atelier de RàPC Grenoble, pp 71-84.

Finnie G., Sun Z., (2003) R<sup>5</sup> model for case-based reasoning, *Know. Based Syst.* 16, pp 59-65.

Lieber J., (2008), Contributions à la conception de systèmes de raisonnement à partir de cas, Mémoire d'Habilitation à diriger des recherches, Université Poincaré, Nancy.

Lieber J., D'Aquin M., Brachais S., Napoli A., (2004) Une étude comparative de quelques travaux sur l'acquisition des connaissances d'adaptation pour le raisonnement à partir de cas. 12ème atelier raisonnement à partir de cas, RàPC'04.

Lopez de Mantaras R., Mc Sherry D., Bridge D., Leake D., Smyth B., Craw S., Faltings B., Lou Maher M., Cox M. T., Forbus K., Keane M., Aamodt A., Watson I., (2006) Retrieval, reuse, revision and retention in case based reasoning, *The Knowledge Eng. Rev.* 20 (3), pp 215-240.

Pralus M., Geneste L., (2007) Recherche et adaptation d'expériences structures, imprécises et incomplètes : application en configuration experte, *Raisonnement à partir de cas 1*, Informatique et Systèmes d'Information, Hermès, Lavoisier, Paris, pp 65-93.

Purvis L., (1998) Synergy and commonality in case based and constraint based reasoning, *AAAI Spring Symposium on Multimodal Reasoning*.