

Université
de Toulouse

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :
Systèmes industriels

Présentée et soutenue par :
Eduardo ROLDAN REYES

le : vendredi 23 novembre 2012

Titre :

Extraction et modélisation de connaissances :
Application à la conception de procédés

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
Laboratoire de Génie Chimique, UMR 5503 - Département Procédés et Systèmes Industriels

Directeur(s) de Thèse :
M. Jean-Marc LE LANN
M. Stéphane NEGNY

Rapporteurs :

M. Jean RENAUD
M. Eric BONJOUR

Membre(s) du jury :

Mme Xuan MEYER, Examineur
M. Nicolas PERRY, Examineur

*A Jesús, mi todo.
A Denisse, mi amor.
A papá y mamá, mi ejemplo.
A Gabo, Fabi y Marquito, mi inspiración.
A la Iglesia de Orizaba y de Toulouse, mis tesoros.*

Remerciements

Tout d'abord je remercie énormément les différents organismes qui ont mis les moyens financiers pour l'aboutissement de cette thèse : à la DGEST, au programme Roberto Rocca et au CONACYT du gouvernement mexicain.

J'adresse mes remerciements aux membres du jury qui ont accepté d'évaluer ce travail. Merci aux professeurs Jean RENAUD et Eric BONJOUR d'avoir accepté d'être les rapporteurs et juger ce travail. Je remercie également les professeurs Nicolas PERRY et Xuan MEYER d'avoir participé en tant qu'examineurs.

Je tiens à remercier Jean Marc LE LANN d'avoir accepté d'être mon directeur de thèse. Un très GRAND merci à mon co-directeur de thèse, Stéphane NEGNY pour son encadrement, ses conseils et de m'avoir guidé pendant ces années de thèse. Merci à Guillermo CORTES ROBLES qui depuis le Mexique, m'a toujours encouragé et nourri de ses expériences mais surtout de son amitié.

Mes plus sincères remerciements à mes collègues du département avec lesquels j'ai partagé de très bons moments : Ali, Anthony, Attia, Benoît, Ferenc, Juliette, László, Marie B., Marie R., Mayra, Moisés, Sayed et Weifeng. Je remercie également Manuel, Guillermo, Jose Luis, Jorge, Ileana et Jorgito, Fer, Liz et Fernandita, Raul, Sofia, Marco et Ale, René et Lety pour votre soutien et amitié pendant mon séjour en France.

J'ai à cœur de remercier toute ma famille et mes amis du Mexique qui malgré la distance ont été toujours là pour m'encourager.

Un grand merci adressé aux familles qui en France, m'ont adopté comme un des leurs : ALMODOVAR, BENNANI, BENQUET, DIAZ, JIMENEZ, MARTINEZ, NAVARRO, PEDROCHE, RADISSON, RATSIMANDRESY, SAUTEL. Vous êtes mon meilleur souvenir de la France.

Mes plus profonds remerciements vont à mes parents dont leur exemple m'a toujours encouragé et guidé. Merci car votre sacrifice et votre amour ont porté leurs fruits dans ma vie. Gabo, Faby et Marquito merci de m'avoir gardé présent tout ce temps dans vos pensées depuis le Mexique. Vous occupez une place importante dans ma vie.

Enfin je remercie du plus profond de mon cœur Denisse, mon épouse, pour toute son aide, soutien et surtout son amour durant mon aventure à Toulouse. Cette réussite n'aurais pas été possible sans toi. Te amo.

Résumé

TITRE : *Extraction et modélisation de connaissances : Application à la conception de procédés.*

L'activité de conception est un processus complexe et décisif dans le cycle de vie des produits et des procédés de fabrication. Dans le contexte actuel, les chercheurs et ingénieurs de conception notent une nette augmentation de la complexité des produits et procédés, pour satisfaire au mieux l'ensemble des exigences croissantes provenant de l'ensemble des acteurs du cycle de vie (industriels et utilisateurs) mais aussi du monde normatif. La gestion des connaissances et de l'expertise métier est un atout important pour rendre plus efficace et accélérer ce processus. Les recherches actuelles sur la gestion des connaissances font émerger des méthodes et outils performants pour identifier, formaliser, exploiter et diffuser la connaissance et les expériences issues de conceptions passées en vue de produire rapidement de nouvelles solutions. Parmi les approches existantes le Raisonnement à Partir de Cas (RàPC) et la Programmation Par Contraintes (PPC) correspondent aux besoins identifiés en Génie des Procédés.

A partir de l'analyse de ces deux approches, ce travail propose un couplage du RàPC et de la PPC afin de fournir un cadre méthodologique et un outil logiciel pour une aide à la conception. Le RàPC permet de capitaliser et de remémorer les expériences passées. Toutefois, la modification de la solution passée pour répondre aux exigences du nouveau problème nécessite l'ajout de nouvelles connaissances aussi appelées connaissances d'adaptation. La PPC, quant à elle, offre justement un cadre approprié pour modéliser et gérer la connaissance permettant l'obtention d'une solution à un problème mais aussi ces connaissances d'adaptation.

Outre la formalisation des connaissances d'adaptation, une des difficultés réside dans l'acquisition de ces connaissances. Dans l'approche proposée, le cycle traditionnel du RàPC a été modifié de façon à créer une boucle d'interaction avec l'utilisateur. Lorsqu'un échec d'adaptation se produit, cette boucle est activée et l'expert est sollicité pour apporter les modifications nécessaires à l'obtention d'une solution appropriée. Cette correction est l'occasion d'acquérir en ligne cette nouvelle connaissance, qui sera par la suite mise à jour et ajoutée dans le système. Un cas d'étude sur la conception d'une opération unitaire de génie des procédés permet d'illustrer l'approche.

MOTS CLES : *Raisonnement à Partir de Cas, Acquisition de Connaissances, Programmation par Contraintes, Adaptation de Cas, Conception de Procédés.*

Abstract

TITLE: *Extraction and Modeling of Knowledge: Application in Process Design.*

Design is a complex and crucial process within the lifecycle of products and production processes. In the current context, design engineers and researchers notice an increasing in complexity of products and processes, in order to meet all the requirements coming from all the participants (manufacturers and users alike) in the life cycle and in the normative world as well. Knowledge management is an important asset to accelerate this process and improve its efficiency. Current research on knowledge management is producing new methods and tools to identify, formalize, exploit and disseminate knowledge from past designs experiences to produce new solutions rapidly. Among existing approaches, Case-Based Reasoning (CBR) and Constraint Programming (CP) are suited to needs identified in Process Engineering.

Based on the analysis of these two approaches, this work proposes a coupling of CBR and the CP to provide a methodological framework and a software tool to assist design. The CBR allows to capitalize and retrieve past experiences. However, transforming the past solution to fit the new problem requirements needs the addition of new knowledge also known as Adaptation Knowledge. CP, meanwhile, offers an appropriate framework to model and manage knowledge required to obtain an appropriate solution to a problem, but also the adaptation knowledge.

In addition to the formalization of adaptation knowledge, one of the remaining major difficulties lies in knowledge acquisition. In the proposed approach, the traditional CBR cycle has been modified to create a user interaction loop. When an adaptation failure occurs, this loop is activated and the expert is asked to make the necessary changes to achieve an appropriate solution. This correction is an opportunity to acquire this new knowledge online, which will be subsequently updated and added into the system. A case study on the design of a unit operation of Process Engineering is used to illustrate the approach.

KEYWORDS: *Case Based Reasoning, Acquired Knowledge, Constraints Programming, Adaptation, Process Design.*

Table de Matières

Introduction générale	1
Chapitre 1 : La gestion des connaissances pour la conception des procédés	6
1 Le Génie des Procédés	6
1.1 Définitions	6
1.2 Spécificité du champ d'application	7
1.3 La diversité des produits	8
1.4 La complexification des systèmes	9
2 Le processus de conception en génie de procédés	11
2.1 Définitions	11
2.2 Les différentes démarches de conception	13
2.2.1 Les démarches orientées produit	15
2.2.2 Les démarches orientées procédé	17
2.3 Problématique	19
3 La gestion des connaissances en conception	20
3.1 Définitions	21
3.2 Les diverses connaissances utilisées en conception	22
3.3 Les approches en gestion de connaissances	23
3.3.1 Approche procédurale	24
3.3.2 Approche axiomatique	24
3.3.3 Les systèmes experts	24
3.3.4 Les approches à partir de cas	25
3.3.5 Les approches à base de contraintes	26
3.3.6 Les approches à base de modèles	27
3.3.7 L'intelligence collective	27
3.4 La gestion des connaissances en génie des procédés	28
4 Le rôle de la modélisation-simulation en génie de procédés	28
4.1 Le modèle mathématique et la capitalisation des connaissances en génie des procédés	29
4.1.1 Développement d'un modèle de procédés	31
4.1.2 Les différents types de modèle en procédés	33
4.1.3 L'évaluation d'un modèle	35

4.2	Les avantages et limites de la modélisation	35
5	Conclusion	36

Chapitre 2 : Le Raisonnement à Partir de Cas 40

1	Introduction.....	40
1.1	Eléments fondamentaux	40
1.2	Le cycle du RàPC	41
1.3	Les connaissances dans le RàPC	43
1.4	Les avantages et les limites du RàPC	45
1.5	Les applications du RàPC.....	46
2	Le RàPC pour le processus de conception	47
2.1	Les avantages du RàPC pour la conception	50
2.2	Le RàPC en conception des procédés.....	50
3	Contexte et travaux précédents	51
3.1	La représentation	51
3.2	Structuration de la base de cas.....	52
3.3	La similarité et ses limites.....	54
3.4	Mesure d'adaptabilité.....	56
3.5	L'exemple d'application du RàPC : le système ReMSiProc.....	58
3.5.1	Description de l'outil ReMSiProc.....	58
3.5.2	Représentation des cas.....	60
3.5.3	Mesure de similarité.....	60
3.5.4	La remémoration dans ReMSiProc	61
3.5.5	L'application de ReMSiProc.....	62
4	Conclusion	64

Chapitre 3 : La Programmation par Contraintes 66

1	Principe général de la programmation par contraintes.....	66
1.1	Applications de la PPC	67
1.2	Les avantages et les limites de la PPC	67
2	Aspects mathématiques des PSC	68
2.1	Les éléments d'un PSC	70
2.2	La modélisation d'un PSC	71
2.3	La résolution d'un PSC.....	73
2.3.1	Les algorithmes de recherche.....	74
2.3.2	Les heuristiques	76

2.3.3	Filtrage sur des domaines discrets	77
2.3.4	Le filtrage sur les domaines continus.....	79
2.3.4.1	L'arithmétique des intervalles.....	80
2.3.4.2	2B-cohérence et Box-cohérence.....	81
3	La PPC en conception	81
3.1	L'intérêt du formalisme PSC en conception.....	82
3.2	La PPC et le processus de conception	83
3.2.1	Le modèle PSC en conception	85
3.3	La programmation par contraintes en procédés.....	86
4	L'exemple d'application de PPC en conception	88
4.1	Le problème de conception d'un échangeur thermique.....	88
4.2	Le modèle du PSC.....	90
4.3	Les solutions.....	92
5	Conclusion	93
Chapitre 4 : Le processus d'adaptation : fondements et problématique.....		96
1	Contexte et principes sur l'adaptation en RàPC	96
1.1	Eléments fondamentaux sur l'adaptation	97
1.2	La formalisation de l'adaptation	97
2	Les travaux sur l'adaptation : différentes approches.....	98
2.1	Les stratégies traditionnelles d'adaptation	99
2.2	L'adaptation par analogie.....	100
2.3	Les approches par décomposition.....	102
2.4	L'adaptation hiérarchique.....	103
2.5	L'adaptation combinatoire ou multi-cas.....	103
2.6	L'adaptation « Machine Learning »	104
2.7	L'adaptation comme un jeu entre espaces	105
2.8	L'adaptation par contraintes.....	105
2.9	L'adaptation interactive	106
3	L'acquisition des connaissances d'adaptation	107
3.1	Les sources des connaissances d'adaptation.....	108
3.1.1	Les sources internes	108
3.1.2	Les sources externes.....	109
3.2	Les formulations des connaissances d'adaptation.....	110
3.2.1	Les règles.....	110
3.2.2	Les opérateurs d'adaptation.....	111

3.2.3	Les cas d'adaptation	111
3.3	Les méthodes utilisées pour acquérir les connaissances d'adaptation	112
3.3.1	Les approches manuelles	112
3.3.2	Les approches automatiques.....	113
3.3.3	Les approches semi-automatiques.....	113
3.3.4	Les approches interactives	114
3.4	Le moment de l'acquisition	114
3.4.1	L'acquisition « Hors ligne »	114
3.4.2	L'acquisition « En ligne ».....	115
4	Positionnement de l'approche proposée d'adaptation	116
5	Conclusion	118

Chapitre 5 : Le couplage du RàPC et de la PPC..... 120

1	Introduction.....	120
2	Les principes du couplage RàPC - PPC.....	120
2.1	Une comparaison entre les deux approches	121
2.2	Les raisons du couplage RàPC - PPC.....	122
2.3	Les limites de l'intégration du RàPC et la PPC.....	123
2.4	Les travaux couplant RàPC et PSC.....	123
2.5	L'identification des différentes modes de couplage.....	128
2.6	L'approche retenue.....	130
3	Le couplage RàPC et PPC en conception	131
3.1	La représentation des cas.....	132
3.2	L'acquisition de connaissances d'adaptation	133
3.3	L'interaction avec l'expert	135
3.3.1	La validation des opérateurs d'adaptation.....	137
3.4	L'évaluation des méthodes d'adaptation.....	138
3.5	La méthodologie du couplage RàPC-PPC	139
4	Conclusion	141

Chapitre 6 : Contexte applicatif : Le système RàP⁴C..... 144

1	Introduction.....	144
2	L'architecture de RàP ⁴ C.....	144
2.1	Le développement de l'outil RàP ⁴ C.....	145
2.1.1	L'accès au système.....	146
2.1.2	La remémoration.....	146

2.1.2.1	La base de cas	147
2.1.3	La réutilisation	147
2.1.3.1	Les méthodes d'adaptations associées	148
2.1.4	La Révision	149
2.1.5	L'Adaptation	149
2.1.5.1	Les modifications de l'utilisateur	151
2.2	Sortie de la session	152
3	Exemple d'application : configuration d'un dispositif agitateur	153
3.1	Introduction	153
3.2	La nature des phases et les opérations	153
3.3	Les caractéristiques hydrodynamiques	154
3.4	Eléments de la configuration d'un mélangeur	155
3.4.1	Les cuves	156
3.4.2	Les chicanes	156
3.4.3	Le dispositif thermique	157
3.4.4	Les mobiles d'agitation	157
3.4.4.1	La position du mobile d'agitation	159
3.4.5	L'entraînement	160
3.4.6	Les arbres	160
3.5	Modèle pour la configuration d'un système d'agitation	161
3.6	Définition du problème pour la configuration d'un système d'agitation	163
3.7	Résolution du problème pour la configuration d'un système d'agitation	164
4	Conclusion	168
	Conclusions générales et perspectives	169
	Bibliographie	175
	Annexe 1 : Algorithme d'indexation sphérique	198
	Annexe 2 : Calcul de la similarité locale	202
	Annexe 3 : Utilisation des sous ensembles flous pour matérialiser l'imprécision	206
	Annexe 4 : Calcul de l'adaptabilité d'un cas	210
	Annexe 5 : Les différents types de mobiles d'agitation	214

Figures et tableaux

Figure 1.1 Les différentes catégories de produits chimiques	9
Figure 1.2 Une classification de la conception.....	14
Figure 1.3 Démarches pour la conception de produits issus des industries de transformation de la matière	17
Figure 1.4 Impact des décisions au travers des différentes étapes de conception	19
Figure 1.5 La transformation de données en connaissances	22
Figure 1.6 La modélisation-simulation d'un procédé.....	29
Figure 1.7 Les différents types de modèle et leurs niveaux de connaissance	34
Figure 1.8 Le modèle de conception préliminaire	37
Figure 2.1 Principe de fonctionnement du Raisonnement à Partir de Cas	41
Figure 2.2 Le cycle R ⁵ du Raisonnement à Partir de Cas	42
Figure 2.3 Comparaison des modèles de conception et du RàPC	48
Figure 2.4 Structure d'un modèle hybride de RàPC en conception	49
Figure 2.5 La représentation de modèles dans ReMSiProc	60
Figure 2.6 L'architecture de ReMSiProc	62
Figure 2.7 Copie d'écran de l'interface graphique de ReMSiProc	63
Figure 2.8 Le modèle de simulation du procédé HDA.....	63
Figure 3.1 Représentation du problème de configuration de réseau	72
Figure 3.2 Algorithme de Backtracking appliqué à la configuration de réseau	75
Figure 3.3 Algorithme de Backtrack intelligent	76
Figure 3.4 Backtrack avec heuristique sur ordre des variables.....	77
Figure 3.5 Algorithme de Forward Checking appliqué à la résolution de l'exemple	79
Figure 3.6 Comparaison du modèle de conception et de l'approche par contraintes.....	84
Figure 3.7 Schéma du refroidisseur d'apéritif	88
Figure 3.8 Solutions en fonction : du nombre d'ailettes (a), du matériau utilisé (b), du choix de diamètres de tubes (c) et de largeurs (d).....	92
Figure 4.1 L'adaptation dans le RàPC.....	97
Figure 4.2 Le carré d'analogie	98
Figure 4.3 Une taxonomie des différentes approches d'adaptation.....	99
Figure 4.4 Adaptation dérivationnelle et transformationnelle	101
Figure 4.5 Le principe du processus d'adaptation en utilisant les PSC	106

Figure 4.6 Une taxonomie des différentes approches pour l'acquisition de connaissances d'adaptation.....	107
Figure 4.7 La boucle d'interaction avec l'utilisateur.....	109
Figure 4.8 Principe du RàPC récursif.....	112
Figure 4.9 L'acquisition hors ligne.....	115
Figure 4.10 L'acquisition en ligne.....	115
Figure 5.1 Modèle d'adaptation de COMPOSER utilisant les PSC.....	124
Figure 5.2 Modèle d'intégration de PSC et RàPC.....	124
Figure 5.3 Modèle RàPC-PSC pour la configuration de produits.....	125
Figure 5.4 Représentation du processus d'adaptation.....	125
Figure 5.5 Modèle RàPC pour la construction de problèmes de PPC.....	126
Figure 5.6 Plateforme générique pour l'adaptation de cas dans le RàPC.....	126
Figure 5.7 Modèle du processus de configuration de produits.....	127
Figure 5.8 Modèle de filtrage à base de cas.....	127
Figure 5.9 Validation du PSC par le RàPC.....	128
Figure 5.10 Validation du RàPC par le PSC.....	128
Figure 5.11 Analyse des cas de la base de cas afin de compléter/modifier le modèle.....	129
Figure 5.12 Utilisation du PSC pour compléter les manquements du RàPC.....	129
Figure 5.13 Réduction du domaine de recherche du RàPC par le PSC.....	129
Figure 5.14 Utilisation du PSC pour la phase d'adaptation du RàPC.....	130
Figure 5.15 Modèle de filtrage à base de cas.....	130
Figure 5.16 Les éléments du couplage RàPC - PPC.....	131
Figure 5.17 Éléments descriptifs d'un cas et de la base de cas.....	132
Figure 5.18 Éléments de la boucle d'interaction avec l'expert.....	134
Figure 5.19 La représentation des connaissances d'adaptation dans un cas.....	136
Figure 5.20 Procédure de test pour les opérateurs d'adaptation.....	137
Figure 5.21 Analyse Pareto pour la qualification des méthodes.....	139
Figure 5.22 Le processus complet d'adaptation.....	140
Figure 6.1 Architecture de RàP ⁴ C.....	145
Figure 6.2 Éléments d'un diagramme de cas d'utilisation.....	145
Figure 6.3 Cas d'utilisation de l'accès du système.....	146
Figure 6.4 Cas d'utilisation de la remémoration.....	146
Figure 6.5 Structure de la base de cas dans RàP ⁴ C.....	147
Figure 6.6 Cas d'utilisation de la Réutilisation.....	148
Figure 6.7 Le système de recommandation dans l'étape de réutilisation.....	148
Figure 6.8 Cas d'utilisation de la solution.....	149

Figure 6.9 Le module d'adaptation	150
Figure 6.10 Cas d'utilisation de la procédure de test des OA.....	150
Figure 6.11 Le cas d'utilisation du module Remémoration.....	152
Figure 6.12 Le dispositif d'agitation	155
Figure 6.13 Les différentes formes de cuves	156
Figure 6.14 Les différentes positions des chicanes	157
Figure 6.15 Cuves munies d'échangeurs de chaleur.....	157
Figure 6.16 Différents schémas d'écoulement produits par les types de mobile d'agitation	158
Figure 6.17 Hauteur du mobile d'agitation	159
Figure 6.18 Positions du mobile d'agitation.....	160
Figure 6.19 Données du problème PPC.....	161
Figure 6.20 Variables et domaines du modèle générique.....	162
Figure 6.21 Exemples de contraintes du modèle Générique.....	162
Tableau 2.1 La typologie des connaissances du RàPC.....	44
Tableau 5.1 Comparaison entre les approches du RàPC et de la PPC.....	122
Tableau 6.1 OA de variables	151
Tableau 6.2 OA de domaines.....	151
Tableau 6.3 OA de contraintes	151
Tableau 6.4 OA de fonction objectif.....	152
Tableau 6.5 Quelques opérations nécessitant de l'agitation	154
Tableau 6.6 Quelques mobiles d'agitation en fonction de l'opération.....	159
Tableau 6.7 Paramètres cinétiques de la synthèse de 1,2 oxyde de propylène	163
Tableau 6.8 OA pour la configuration d'un dispositif agitateur	166
Tableau 6.9 Les six configurations résultantes du dispositif mélangeur.....	167

Introduction générale

La conception est une activité décisive sur l'ensemble du cycle de vie d'un produit ou d'un procédé. C'est à travers elle que les solutions à mettre en œuvre sont déterminées afin de réaliser un système conforme aux exigences. Face à un développement toujours plus rapide et fluctuant dans un environnement de plus en plus complexe, les entreprises cherchent à prendre un avantage concurrentiel [BON 07]. Elles se penchent pour la capitalisation de leurs connaissances et expertises afin de réduire les coûts, améliorer les fonctionnalités, accroître la qualité. En effet, il est fréquent qu'un concepteur s'appuie sur des expériences passées pour commencer une nouvelle session de conception.

De nombreuses décisions sont arrêtées durant cette activité. Remettre en cause l'un des choix génère un allongement de l'activité. Il est donc primordial de réaliser des choix judicieux dès le début du processus pour éviter d'itérer et pour converger plus rapidement vers une solution. L'exploitation de l'expérience acquise permet de réduire fortement les délais puisque certains choix ne sont ni à faire, ni à remettre en cause. Dans ce contexte, il est souhaitable de disposer d'approches et d'outils d'aide à la conception exploitant les connaissances et savoirs-faire. La mise en place de ces outils passe par différentes étapes telles que la formalisation et l'exploitation des connaissances, ce qui présuppose une certaine récurrence dans l'activité.

L'approche à base de modèles mathématiques est la plus communément utilisée en génie des procédés pour capitaliser les connaissances. Cependant cette capitalisation n'est que partielle et ne permet pas de modéliser l'ensemble des connaissances et plus particulièrement la connaissance tacite. De plus, en conception préliminaire, il faut également capitaliser une expertise autre que celle traditionnellement visée par ces modèles, notamment celle sur les choix technologiques et de configuration de systèmes. Des approches issues du champ de l'Intelligence Artificielle, apportent des réponses potentielles pour la gestion des connaissances déployées en conception préliminaire des procédés. Elles ont donné lieu à des résultats probants dans ce domaine d'application mais également d'autres. Parmi celles-ci, le Raisonnement à Partir de Cas (RàPC) et la Programmation par Contraintes (PPC) émergent comme des méthodes adaptées pour capitaliser et diffuser les expériences de conception afin de construire une mémoire pour produire des solutions répondant aux exigences du problème.

Ce travail s'inscrit dans une suite de travaux visant à mettre à disposition des concepteurs un cadre méthodologique et un outil facilitant la résolution de problèmes de conception en Génie des Procédés. Ces différents travaux menés dans le département Procédés et Systèmes Industriels du Laboratoire de Génie Chimique se focalisent sur la gestion des connaissances. Pour compléter l'approche par modélisation (à l'origine du département), l'utilisation de techniques de RàPC semble assez naturelle pour réaliser les traitements désirés. Les travaux antérieurs ont cherché à adapter certaines étapes du RàPC d'une part à la conception, d'autre part au Génie des Procédés mais aussi à améliorer certaines phases du RàPC. Ainsi, comme illustré sur la figure A.1, cette thèse s'inscrit dans la continuité de travaux sur :

- la représentation : prise en compte de l'incertitude et l'incomplétude des données [NEG 10] ;
- la similarité : introduction de connaissance spécifique du domaine dans le calcul de mesures de similarité [N&L 08A] ;
- la structuration de la base de cas : mise en œuvre d'un algorithme rapide et efficace basé sur une indexation sphérique [NEG 10] ;
- l'adaptabilité : anticipation de la phase d'adaptation dès l'étape de remémoration pour la faciliter. Mise en place d'un calcul de l'adaptabilité d'un cas par utilisation des ensembles flous [N&L 08B].

Cette thèse s'intéresse plus particulièrement à la phase d'adaptation du RàPC. La finalité est de proposer une méthodologie et un outil pour réaliser efficacement cette phase grâce à un couplage entre le RàPC et la PPC, figure A.1.

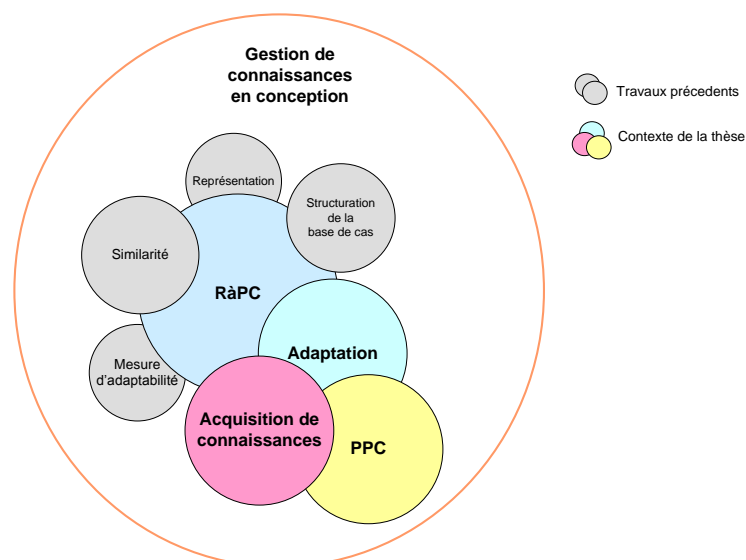


Figure A.1 Positionnement de la thèse

Organisation du manuscrit

Une synergie entre la méthode de capitalisation de connaissances RàPC et le paradigme de programmation par contraintes PPC est présenté dans ce rapport. Ce document est organisé en six chapitres (schématisés sur la figure A.2 en fin de section) :

Le premier chapitre a pour objectif d'offrir un aperçu sur la conception dans le domaine des procédés. Les différentes approches de la conception ainsi que certains modèles pour représenter ce processus sont discutés. Les concepts de la gestion de connaissances, les différentes approches d'exploitation, et leurs rôles dans le processus de conception sont introduits, et notamment le RàPC et la PPC.

Les bases sur le RàPC sont décrites dans le chapitre 2. Il est abordé comme une méthodologie [WAT 99] permettant de résoudre des problèmes en suivant un principe général : des problèmes similaires ont des solutions similaires. Dans ce cadre, le RàPC cherche dans sa base de cas un problème similaire et réutilise sa solution pour résoudre le problème rencontré. Puis ce dernier avec la nouvelle solution devient une nouvelle expérience qui peut être stockée dans la base de cas. Un exemple d'un système RàPC appliqué à la simulation des procédés est proposé à la fin du chapitre.

La programmation par contraintes est expliquée dans le chapitre 3. Elle est présentée comme une approche permettant la résolution de problèmes complexes de nature combinatoire. Dans ce chapitre on présente un aperçu sur cette technique mathématique et son apport pour la conception. Le but est de l'intégrer au RàPC pour améliorer l'étape d'adaptation de ce dernier. Un exemple d'application est également proposé.

Le chapitre 4 décrit l'étape d'adaptation, primordiale pour rendre un système de RàPC efficace. Conceptuellement simple, cette étape est d'une concrétisation complexe. En effet, elle demande des connaissances supplémentaires et spécifiques qui génèrent des problèmes d'acquisition, formalisation et de gestion. Ce chapitre aborde cette problématique, tout d'abord en détaillant ses principes généraux, ensuite ses enjeux et enfin l'approche retenue pour la suite de ce travail.

La description de l'implémentation du couplage RàPC-PPC est présentée dans le chapitre 5. Ce couplage a pour but de faciliter l'étape cruciale d'adaptation du RàPC qui demande un effort d'ingénierie important en particulier pour la représentation et

l'acquisition des connaissances d'adaptation. Pour leur représentation la PPC est utilisée. Pour leur acquisition il est intéressant de disposer d'un moyen de tirer profit des épisodes de résolution. Pour ce faire une boucle d'interaction avec l'expert est mise en œuvre pour acquérir les connaissances au fur et à mesure des résolutions de problèmes sans contraindre l'expert.

Le dernier chapitre présente l'application de la démarche au travers d'une étude de cas. Elle est mise en œuvre dans le système prototype RàP⁴C. Puis un exemple est proposé, sur la conception d'un système d'agitation. Il montre la façon dont un problème de conception est représenté et recherché dans la base de cas, puis un modèle de PPC est traité par le module d'adaptation pour calculer une solution satisfaisant les exigences du problème.

Finalement, les conclusions et perspectives générales sur l'approche présentée sont discutées.

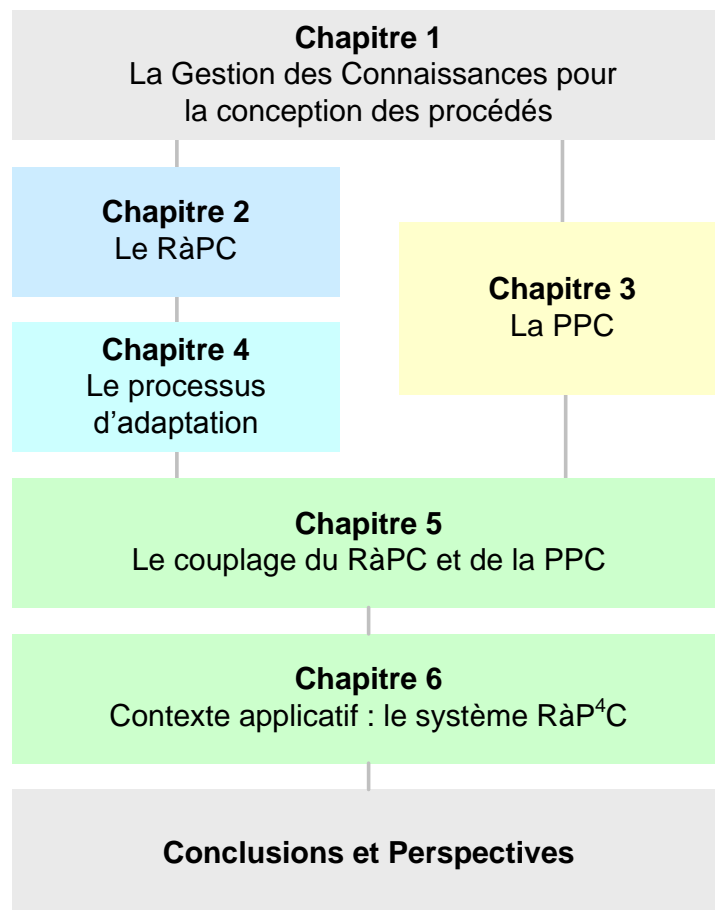


Figure A.2 Structure du manuscrit

Chapitre 1

La gestion des connaissances pour la conception des procédés

Ce chapitre a pour objectif d'offrir un aperçu sur la conception dans le domaine du génie des procédés. On y discute les différentes approches de la conception ainsi que certains types de modèles pour représenter ce processus. On introduit les concepts de gestion des connaissances, les différentes approches d'exploitation, et leurs rôles dans le processus de conception.

Chapitre 1 : La gestion des connaissances pour la conception des procédés

1 Le Génie des Procédés

Le génie des procédés englobe un ensemble de connaissances et de méthodes indispensables pour la conception et l'exploitation optimale des procédés dans plusieurs domaines liés à la transformation de la matière et de l'énergie. Au cœur de ces deux dernières activités, il est la partie de l'ingénierie qui a pour but de convertir des matières premières, en des produits avec une plus forte valeur ajoutée grâce à des transformations de la matière et de l'énergie sous des contraintes économiques, environnementales et de sécurité. Au niveau du laboratoire, le chimiste s'intéresse aux mécanismes réactionnels à l'échelle de la liaison chimique et des structures moléculaires. L'ingénieur de génie des procédés en revanche, définit et dimensionne les installations où se déroulent les réactions chimiques et leurs conditions d'exploitation à l'échelle industrielle.

1.1 Définitions

Une première définition récente [ENS 10] définit le génie des procédés par : *«...l'ensemble des connaissances scientifiques et technologiques nécessaires à la conception et à la conduite des procédés de transformations physiques, chimiques ou biologiques de la matière et de l'énergie »*. Une définition plus précise a été proposée par la Société Française de Génie des Procédés [SFG 10] : *« Science des systèmes complexes qui recouvre l'ensemble des connaissances et des méthodes nécessaires à la conception et à la mise en œuvre des procédés industriels de transformation de la matière et de l'énergie »*. Ces deux définitions mettent clairement en évidence l'importance du processus de conception dans la transformation de la matière et de l'énergie. Elles décrivent la conception comme la base primordiale à mettre en œuvre pour tout procédé allant jusqu'à l'obtention d'un produit. Les domaines concernés rassemblent aussi bien les procédés chimiques, qu'agroalimentaires, ou biologiques. A l'heure actuelle les aspects environnementaux tels que le traitement de l'eau, la gestion des déchets ou les énergies renouvelables, se situent en tête des priorités des entreprises du domaine.

1.2 Spécificité du champ d'application

En génie des procédés, un projet de conception nécessite souvent de mener conjointement les étapes de développement du produit et le développement du procédé de fabrication. Dans cette discipline, le couple produit/procédé est fortement lié car les méthodes et les conditions opératoires utilisées au niveau du laboratoire sont rarement adaptées à la production de masse, notamment pour des raisons économiques et techniques. Par conséquent, la conception du procédé de fabrication peut influencer le mode d'obtention du produit : choix des matières premières, choix des voies de synthèse, choix des techniques de séparation, choix des conditions opératoires.

Lorsque le plan de synthèse (assimilable à la conception produit) a atteint un certain degré d'avancement, les informations nécessaires (conditions opératoires, données cinétiques) sont transmises à l'ingénieur procédé pour la conception de l'atelier de fabrication. La conception de procédés chimiques nécessite la sélection d'une série d'étapes de traitement de la matière et de leur intégration pour former un système complet de fabrication, résumé sous forme d'un schéma communément nommé « *flowsheet* ». Sur ce flowsheet figurent les principaux appareils, les conditions opératoires accompagnées des bilans (matière et énergie), les boucles de contrôle et régulation. Il faut donc définir les différentes opérations unitaires¹, choisir le réacteur, les agencer (choix du train de séparation), dimensionner les divers appareillages, optimiser le procédé et enfin envisager son contrôle et sa commande. La conception du procédé est complexe de par les phénomènes physiques mis en jeu, mais aussi de par la prise en compte d'une grande variété de contraintes traitées progressivement. A cause du temps et du coût de développement d'un nouveau procédé chimique d'une part, et du lien produit/procédé d'autre part, une approche par ingénierie simultanée est régulièrement utilisée.

En termes économiques, l'importance de la conception de procédés devient évidente lorsqu'on réalise que, bien qu'une fraction relativement petite de tout le budget soit dépensée pendant cette étape, la plupart des coûts y sont assignés [SEU 06]. Douglas dans [DOU 88] a estimé que l'étape de conception de nouveaux procédés représente 10% voire 20 % du coût total, mais les décisions prises durant cette phase représentent 80 % de tous les coûts du projet. [SEU 06] note qu'en conception de procédés un grand nombre de concepts et de solutions potentielles sont générés ce qui nécessite une phase d'évaluation rapide et à moindre coût. Selon lui, la difficulté principale avec ce grand nombre d'alternatives de solutions est que le chemin entre la prise de décisions de conception et les demandes à

¹ Une subdivision d'un procédé industriel qui consiste en une transformation physique, chimique ou biologique.

satisfaisante peut s'avérer complexe et non linéaire. Ceci exige donc des méthodes systématiques qui réduisent le nombre d'alternatives de solutions le plus tôt possible dans le processus de conception.

1.3 La diversité des produits

L'industrie des procédés est très diversifiée : pétrochimie, chimie fine, nucléaire, agroalimentaire... De nombreux produits chimiques et dérivés sont fabriqués par les industries du procédé, le spectre des produits issus de ces industries est relativement large. De l'extraction des matières premières à la fabrication du produit final, les éléments constitutifs de ce dernier ont la plupart du temps subi de nombreuses modifications dans plusieurs procédés ou étapes intermédiaires. Cette industrie comprend des industries de base permettant soit à des industries du secteur soit à d'autres industries de disposer de produits intermédiaires dont elles ont besoin, mais aussi des industries de produits finis dédiés à la grande consommation. Il est donc possible de classer les produits en trois catégories, suivant que le produit est plus ou moins en amont dans le circuit industriel, donc plus ou moins proche du consommateur, figure 1.1 : les produits chimiques de base, les produits industriels, produits dédiés au marché de grande consommation. Les premiers sont produits à partir de ressources naturelles pour la fabrication de produits chimiques intermédiaires de base ou de spécialité, comme par exemple ; l'éthylène, l'acétone, le chlorure de vinyle... Les polymères font également partis de cette catégorie qui englobe des molécules ou mélanges bien définis et connus mais rarement vendus directement aux consommateurs. Ces produits servent ensuite de matière première pour élaborer des produits industriels : films, fibres, pâtes, crèmes... Bien qu'ils soient caractérisés par des propriétés thermodynamiques, physiques et de transfert, il est souvent nécessaire de leur adjoindre des propriétés plus spécifiques notamment sur leur structure moléculaire et leur composition afin de satisfaire les exigences des consommateurs. Par exemple les crèmes et pâtes sont des colloïdes avec des microstructures et des distributions de taille de particules caractérisées par des propriétés fonctionnelles (pouvoir nettoyant, adhésion), sensorielles, rhéologiques, physiques (stabilité). Tout comme les précédents, ces produits industriels sont majoritairement des intermédiaires intervenant dans la réalisation de produits dédiés à la grande consommation : cartouches d'encre, médicaments, cosmétiques, détergents. Les crèmes et pâtes précédentes sont mélangées avec d'autres composés pour former les cosmétiques avec les propriétés désirées puis conditionnées.

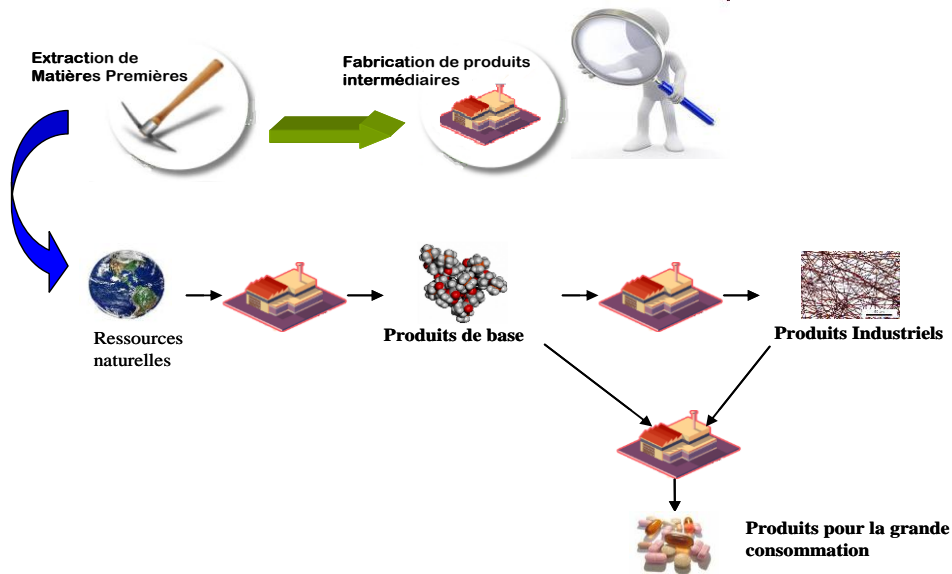


Figure 1.1 Les différentes catégories de produits chimiques

1.4 La complexification des systèmes

Dans tous les domaines techniques, les chercheurs et ingénieurs de conception notent une nette augmentation de la complexité des systèmes, pour satisfaire au mieux l'ensemble des exigences croissantes provenant de l'ensemble des acteurs du cycle de vie mais aussi du monde normatif. Il devient difficile de trouver une solution répondant à l'ensemble de ces exigences, surtout que ces dernières sont parfois contradictoires voire antagonistes. Il faut ajouter de plus en plus de fonctionnalités nécessitant l'intégration de nouveaux composés, composants ou sous-systèmes. Cette intégration est souvent un compromis entre l'augmentation de performance due à cet ajout et les contraintes liées aux nouveaux défis à relever comme celui du développement de produits plus légers, plus performants, plus sûrs, plus durables, plus respectueux de l'environnement, etc.

Bien que les premières étapes de la conception cherchent à clarifier et lister l'ensemble des exigences, il est très rare qu'elles soient totalement connues dès le départ. Des évolutions imprévues, le manque d'information, des besoins client mal perçus, mal spécifiés ou qui évoluent, peuvent venir perturber les différentes étapes de la conception. Cette incertitude, cette ambiguïté et ce caractère incomplet sur sa formulation accroissent la complexité du problème.

À la complexification des produits s'ajoute la complexification de la technologie et plus largement des procédés, pour répondre aux nouvelles exigences des produits d'une part et

aux exigences de production actuelles (environnementales, énergétiques, économiques, sécurité, flexibilité) d'autre part. Toutes les évolutions technologiques liées à l'intensification des procédés en sont un exemple illustratif. Au cours du temps, les transformations continues du produit sont aussi une des causes de sa complexification grandissante.

De part cette complexification, les chercheurs doivent élargir leur domaine de connaissances pour traiter tous les aspects techniques liés au produit et au procédé. Cet élargissement des connaissances est d'autant plus nécessaire lorsqu'on aborde l'innovation. La génération d'idées nouvelles et de concepts innovants fait souvent appel à des connaissances techniques d'autres domaines scientifiques que l'on doit identifier puis adapter plus spécifiquement à son domaine puis à son problème de conception (périmètre métier). D'un côté les chercheurs doivent posséder des compétences et connaissances pointues dans leur domaine, de l'autre ils doivent avoir un large champ de connaissances pour proposer des innovations. Cette contradiction en termes de compétences humaines peut être résolue :

- soit par la mise en place d'une équipe pluridisciplinaire avec toutes les difficultés que cela génèrent. Dans les centres de R&D des grands groupes chimiques, les équipes sont composées de : chimistes, ingénieurs de génie chimique, biologistes, physiciens, mécaniciens, ingénieurs matériaux, automaticiens... avec parfois comme difficulté supplémentaire, la répartition des compétences dans divers lieux géographiques ;
- soit par la mise en œuvre d'approches et/ou de nouvelles méthodes de conception transdisciplinaires donnant accès à un large spectre de connaissances.

Actuellement, le génie des procédés est au cœur d'un changement technologique, l'économie se trouvant remodelée par les nouvelles technologies de l'information et par des transformations radicales dans des domaines comme les biotechnologies, la science des matériaux, l'intensification des procédés, les sciences du vivant, les contraintes sociétales. Les vingt priorités du génie des procédés identifiées par l'Institution of Chemical Engineering (IChemE), regroupées en six thèmes (Sustainability and Sustainable Chemical Technology, Health Safety Environment and Public perception of Risk, Energy, Food and Drink, Water, Bioprocess and Biosystems Engineering), donnent une idée et une vision des enjeux futurs pour cette industrie.

2 Le processus de conception en génie de procédés

La conception est une activité centrale en génie des procédés au même titre que dans de nombreuses autres disciplines. Par conséquent, l'activité de conception a suscité de nombreux travaux de recherche. Après analyse des différents processus de conception élaborés dans les diverses disciplines, on retrouve des éléments communs indépendants du domaine d'application, dans la démarche de conception et dans la façon d'en aborder les objectifs. La conception commence à partir de la formulation d'une idée, ou à partir de l'expression d'un besoin en vue d'une amélioration des fonctionnalités d'un système existant ou de la création d'un artefact avec de nouvelles fonctions. L'activité de conception doit commencer avec la définition d'objectifs, de contraintes selon lesquelles les objectifs doivent être atteints, avec des critères de mesure d'atteinte de ces derniers. Le résultat de l'activité est une description détaillée d'un artefact fourni par un ensemble de spécifications.

2.1 Définitions

Plusieurs ingénieurs, chercheurs et concepteurs ont exprimé, à partir de leurs expériences, différents points de vue en formulant leur propre définition de la conception. Certains de ces points de vue sont résumés par la suite :

- Dans le sens, le plus classique, la conception est selon Feilden [FEI 63] « *l'utilisation de principes scientifiques, informations techniques et de l'imagination lors de la définition d'une structure mécanique, machine ou système pour réaliser des fonctions spécifiques en maximisant l'efficience et un critère économique* ».
- La conception est « *le processus créatif qui commence avec certaines conditions et définit un système et les méthodes pour sa réalisation ou implémentation en respectant ces exigences* » [F&F 83].
- Luckman [LUC 84] à son tour définit la conception comme « *la transformation de l'information exprimée sous la forme de conditions, de contraintes et d'expériences, en solutions potentielles qui satisfassent les besoins pour lesquels la transformation a été faite ... à ceci doivent s'ajouter la créativité et l'originalité pour que ce soit appelé conception* ».

Un point commun aux définitions précédentes, est que toutes soulignent la conception comme étant motivée par un ensemble de besoins exprimés sous différentes formes. Ces auteurs, se basent sur différents concepts tels que les spécifications, les contraintes, la créativité ou les solutions pour définir la conception. En intégrant tous ces mots clés,

Evbuomwan [EVB 96] définit la conception comme : « *Le processus qui cherche à établir des conditions basées sur les besoins humains, les transformant en spécifications de performances et fonctions, qui sont alors modélisées et converties (soumises aux contraintes) en solutions de conception (en utilisant la créativité, les principes scientifiques et les connaissances techniques) qui peuvent ensuite être matérialisées avec un bénéfice économique.* »

Durant le processus de conception, plusieurs activités et propriétés sont en jeu. C'est encore Evbuomwan [EVB 96] qui en énumère quelques-unes :

1. La conception comme une activité opportuniste, représente le cas où les approches « top-down² » et « bottom-up³ » sont utilisées par le concepteur d'une manière opportuniste.
2. La conception comme une activité incrémentale, comporte un processus évolutionnaire où des changements sont proposés afin d'atteindre une meilleure conception.
3. La conception comme une activité exploratoire, implique un modèle basé sur des explorations de conception et décrit le processus de conception comme une tâche basée sur des connaissances.
4. La conception comme un processus de recherche, englobe une analyse des besoins du client, des techniques de conception disponibles, des solutions précédentes de conception qui soient semblables, échecs et succès passés entre autres.
5. La conception comme un processus créatif, comprend la création d'une solution avec l'aide du savoir-faire, de l'ingéniosité, de la mémoire, des capacités de reconnaissance de formes, des recherches aléatoires dans l'espace des solutions, des analogies...
6. La conception comme un processus rationnel qui porte sur la vérification et le test des solutions proposées, impliquant le raisonnement logique, l'analyse mathématique, la simulation sur ordinateur, les résultats d'expériences dans les laboratoires...
7. La conception comme un processus de prise de décision où les concepteurs doivent faire des choix en adoptant des lignes d'action alternatives ou en choisissant parmi des solutions de conception concurrentes. De telles évaluations sont basées sur l'expérience ou des critères extraits à partir des besoins du client.
8. La conception comme un processus itératif. L'activité itérative est le processus le plus commun dans la conception. Des conceptions préliminaires sont habituellement analysées avec les contraintes actuelles, si elles ne les satisfont pas, une correction est conduite basée sur l'expérience et les résultats de l'analyse.
9. La conception comme un processus interactif introduit le concepteur directement au cœur du processus en le forçant à en être partie intégrante. Ceci est rendu nécessaire dans les situations où : a) le problème de conception est mal défini, b) il n'y a pas d'outils

² L'approche « top-down » formule un résumé du système, sans en préciser les détails.

³ L'approche « bottom-up » conçoit les parties individuelles en détail pour être ensuite liées au système.

analytiques développés pour permettre l'analyse quantitative et c) il y a peu ou pas d'expérience disponible ou associée au problème de conception.

Toutes les activités décrites précédemment représentent différentes parties de la globalité du processus de conception [EVB 96]. Elles dépendent de l'expertise des concepteurs ainsi que du domaine d'application. La plupart sont complémentaires et il est souhaitable que la conception s'appuie largement sur plusieurs d'entre elles.

A ce stade, l'activité de conception a été définie de plusieurs manières et sous différents angles. Vernat [VER 04] dans sa recherche affirme « *Il semble difficile de définir l'activité de conception en ingénierie sans être restrictif, tant elle est complexe et tant elle peut recouvrir des champs divers* ». C'est pourquoi plusieurs auteurs ont proposé de nombreuses descriptions du déroulement de l'activité de conception issues aussi bien du monde industriel ou normatif que du monde académique. Ces différentes démarches sont présentées dans le point suivant.

2.2 Les différentes démarches de conception

Il existe de nombreuses descriptions et de nombreux formalismes du déroulement de l'activité de conception. Scaravetti [SCA 04] a examiné ces différentes démarches ainsi que les différents modèles associés, pour en dégager les caractéristiques communes mais également leurs spécificités. Parmi toutes ces démarches de référence, il est à noter les travaux les plus communément cités à savoir : la conception axiomatique [SUH 90], l'approche systématique et séquentielle de Pahl et Beitz [P&B 96], ainsi que les travaux d'Ulrich et Eppinger [U&E 00] sur l'approche modulaire.

Qu'il s'agisse de conception séquentielle ou simultanée, il existe un découpage qui s'articule autour des mêmes phases (provenant du besoin de segmenter le temps pour prévoir des étapes de prises de décision) : définition du problème qui aboutit à la spécification des exigences, la conception préliminaire (proposition de concepts et mise en place de structures fonctionnelles) et la conception détaillée (description complète). A noter que cette dernière phase est souvent assimilée à la conception tout court [YAN 01], ce qui est souvent le cas en Génie des Procédés.

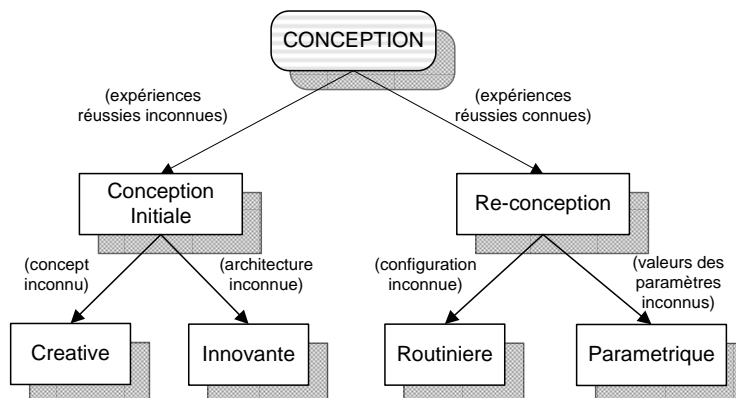
Une autre classification, faite il y a quelques années par [EVB 96], distinguait deux types de modèles, les prescriptifs (orientés sur des méthodologies et procédures) et les

descriptifs (focalisés sur les actions et activités du concepteur). Elle envisageait déjà un nouveau groupe de modèles basés sur l'emploi de techniques informatiques numériques et qualitatives, ainsi que des modèles reposant sur des techniques d'intelligence artificielle.

Plusieurs approches de conception sont basées sur une modélisation qualitative, provenant du domaine de l'intelligence artificielle [Z&R 10]. Une des premières approches est celle proposée par Williams [WIL 92], basée sur l'idée que les premières étapes d'une nouvelle conception identifient toutes les interactions existantes entre les composants qui constituent le problème. Ces interactions sont créées grâce à la formulation de certains principes qui sont essentiellement des lois physiques. Toutefois, il existe une difficulté majeure lors de la génération de toutes les interactions possibles, car le problème peut atteindre une dimension difficile à modéliser.

D'autres modèles, comme celui proposé par Tomiyama [TOM 02] sont basés sur une combinaison d'une structure fonctionnelle et une modélisation qualitative. Chakrabarti [CHA 02] envisage le problème de la conception en proposant un modèle qui manipule un grand nombre de solutions stockées dans une mémoire. On doit considérer que ces modèles sont habituellement prévus pour être traités par des outils informatiques en utilisant de plus grandes bases de données et pourraient être incertains sans mécanismes appropriés pour la gestion d'une éventuelle explosion combinatoire et d'une analyse empirique.

Les problèmes de type combinatoire sont résolus par des méthodes (proposées par Ziv-Av et Reich [Z&R 05]) de modélisation de contraintes sous une approche d'exclusivité mutuelle et de nécessité fonctionnelle, qui permettent de résoudre un problème de type optimisation. Puis Chong [CHO 09] propose l'intervention d'un utilisateur qui peut explorer et gérer les concepts dans tout l'espace de la conception du produit.



Source : [SER 06]

Figure 1.2 Une classification de la conception

Dans une autre approche [SER 06] décline l'activité de conception en deux classes pour en donner quatre types. La figure 1.2 ci-dessus présente cette classification. Ces deux classes sont la conception initiale qui traduit une absence de connaissances voire d'expériences sur le domaine traité et la re-conception qui est plutôt adaptée à l'ingénierie à base de connaissances, puisqu'elle s'appuie sur des éléments développés auparavant [MON 08].

Dans sa recherche, Seuranen [SEU 06] classe les différentes approches du processus de conception en génie des procédés en trois catégories :

- les méthodes basées sur des heuristiques et des expériences d'ingénierie qui utilisent une approche plutôt hiérarchique, comme le modèle « oignon » dans [SMI 05],
- les approches d'optimisation utilisant par exemple des modèles MINLP (mixed integer non-linear programming) [GRO 02], des algorithmes génétiques pour résoudre des problèmes multicritères [KHA 00] ou le « recuit simulé » proposé dans [FAB 05],
- les méthodes basées sur des connaissances, comme l'approche basée sur les phénomènes pour la génération d'alternatives de procédés intensifiées [RON 04], la méthode CBA (Conflict-Based Approach) utilisant l'approche de TRIZ pour la modification de l'espace de solutions afin de proposer des alternatives durant la phase préliminaire de conception [LIX 02], l'approche évaluative basée sur la combinaison de meilleures propriétés des différentes alternatives des procédés existants [CZI 05] et le RàPC pour remémorer des cas de conception, les réutiliser et les adapter pour résoudre de nouveaux problèmes de conception (voir au chapitre 3 pour des exemples d'application) .

Dans les sections suivantes, une synthèse des différents travaux, issus particulièrement du génie des procédés, est présentée.

2.2.1 Les démarches orientées produit

L'identification des propriétés et des fonctionnalités d'un produit chimique qui correspondent aux besoins, et leurs transitions vers des propriétés structurales du produit peuvent être considérées comme la conception produit en génie chimique (« Product Engineering »). Pendant cette phase, les ingénieurs de génie chimique font face à deux tâches principales [A&K 08] : la détermination de la composition du mélange chimique ou de la structure du nouveau composé chimique dans le but d'atteindre les propriétés désirées, et la création du flowsheet du procédé avec les conditions de fonctionnement pour produire les produits voulus avec un degré élevé de rendement et de sélectivité.

Une démarche générale pour la conception des procédés a été suggérée par Seider [SEI 04]. Lorsqu'on compare cette démarche avec celle de Pahl et Beitz [P&B 96] qui fait référence pour la conception de produits complexes issus de l'industrie manufacturière, on constate également un déroulement progressif de l'activité de conception autour de quatre phases. Il est à noter que la démarche de Pahl et Beitz est fortement centrée sur le produit, l'aspect procédé de fabrication n'apparaissant que dans la dernière phase de conception détaillée, tandis que l'autre se focalise sur le procédé avec la prise en compte du produit que dans sa phase initiale. Malgré cette divergence, ces deux démarches mettent en évidence la relation produit/procédé même si de nombreux chercheurs différencient ces activités. Même si elle n'est pas complètement adaptée pour les procédés chimiques, la démarche de Pahl et Beitz peut s'appliquer à la conception des appareillages technologiques d'un procédé comme par exemple les réacteurs, les échangeurs de matière... qui s'apparentent ainsi à des produits de l'industrie manufacturière.

Certains auteurs ont complété la démarche de Seider [SEI 04] en précisant la première étape et plus particulièrement la conception de produit chimique. Certes il est possible de transposer certaines démarches centrées sur un produit manufacturier, mais il est souvent nécessaire de les adapter aux produits chimiques. Wibowo et Ng [W&N 01] décrivent un modèle pour la conception de produit chimique avec une application pour la formulation de cosmétiques (crèmes et pâtes). Cette approche illustrée sur la figure 1.3a se décompose en quatre étapes principales dont les deux premières se focalisent sur le produit. A noter que cette démarche s'accompagne de règles heuristiques pour spécifier les facteurs sensoriels (étape 1) et pour la formulation des produits (étape 2).

De leur côté, Cussler et Moggridge [C&M 01] ont également proposé un modèle à étapes pour la conception de produit chimique, figure 1.3b. Chaque étape se décompose en sous tâches à accomplir. Après l'analyse du besoin et sa traduction sous forme de spécifications, l'étape suivante cherche à générer des concepts à partir de différentes sources puis elle vise à les trier, à les évaluer pour finalement retenir que les plus pertinents et prometteurs. Il s'ensuit une étape de sélection plus sévère qui se base sur des critères physiques sur le produit : la thermodynamique, les cinétiques, mais aussi une étude de faisabilité. La dernière étape est essentielle afin de fournir une description complète et rigoureuse des concepts retenus. Rechercher les détails sur les concepts pourrait nécessiter de nouvelles recherches et expérimentations (notamment sur la voie de synthèse de la molécule désirée). C'est à partir de cette étape que commence la conception du procédé de fabrication.

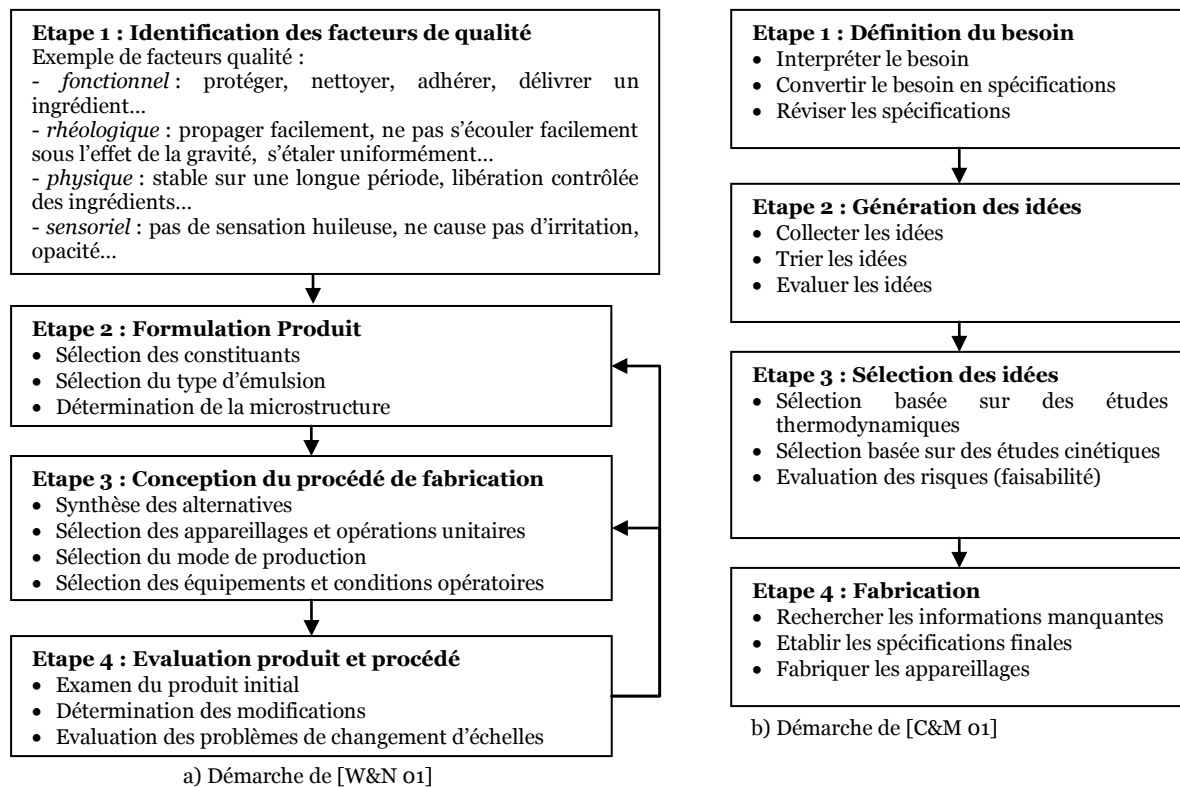


Figure 1.3 Démarches pour la conception de produits issus des industries de transformation de la matière

2.2.2 Les démarches orientées procédé

Avec le développement produit se pose automatiquement la question de la conception de son procédé de fabrication. La conception de ce dernier établit le choix et la séquences des opérations chimiques et physiques, les conditions opératoires, les fonctions, configurations et spécifications de tous les équipements, la disposition des équipements pour assurer un bon fonctionnement de l'atelier de production. Le résultat de la conception du procédé est résumée par un schéma de procédé (flowsheet), des bilans matières et énergie ainsi qu'un ensemble de spécifications sur les équipements individuels.

Dans [P&B 96], les aspects liés au procédé de fabrication sont traités très en aval, dans l'étape de conception détaillée. Pour le génie des procédés et pour certaines entreprises manufacturières, les interactions produits/procédés sont telles qu'il n'est pas envisageable de se préoccuper si tardivement du moyen de production. [BIE 97] propose un modèle qui divise le processus de synthèse en plusieurs tâches secondaires. Après la description des spécificités d'un problème dans l'étape de génération de concepts, plusieurs approches, pour traiter la

conception, sont identifiées. Ensuite, lors de la génération d'alternatives de solutions, les informations sont analysées de manière approfondie pour trouver des procédés existants ou pour en développer de nouveaux. Dans l'étape d'analyse chaque procédé est analysé en calculant quelques paramètres : des bilans énergétiques, des températures ou des pressions... Ensuite dans l'étape suivante, la performance du procédé est évaluée. Finalement, l'étape d'optimisation permet d'améliorer le procédé en ajustant et en affinant les décisions prises dans les étapes précédentes.

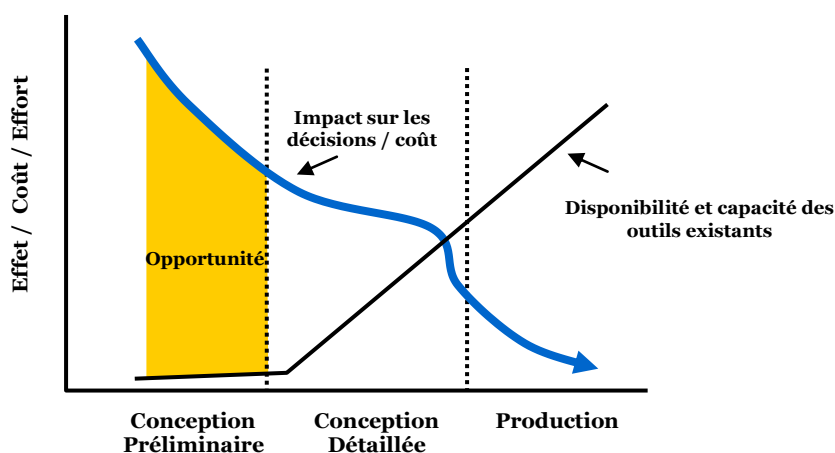
Dans son approche générale, Cooper [COO 05] a proposé une démarche de conception intégrant les aspects produits, procédés et technologiques dès les premières phases de la conception. Ce processus est couramment employé dans les industries, après avoir été adaptés aux besoins et à l'environnement de l'entreprise. Il est composé de plusieurs étapes se concluant de jalons de décision regroupant les parties prenantes et les différents décisionnaires (techniques, production, logistique, sécurité). A chaque jalon, les décisionnaires peuvent soit passer à l'étape suivante, soit arrêter momentanément le projet en attendant que les problèmes critiques soient résolus, soit arrêter complètement le projet.

Bien que cette démarche reprenne les mêmes éléments que les précédentes, elle possède l'avantage d'introduire la conception du procédé de fabrication dans les étapes amont du développement produit afin de diminuer le temps de conception. Par rapport à la démarche de Seider [SEI 04], elle possède également l'avantage d'intégrer une réflexion sur la technologie dès les premières phases de la conception produit. La première étape a pour objectif de réaliser une veille technologique et scientifique, de générer des idées et d'évaluer les alternatives pour la conduite du projet de développement technologique. L'étape suivante est plus étendue, son objectif est de démontrer la faisabilité technologique, c'est-à-dire de montrer que la nouvelle technologie fonctionne correctement et mérite une plus grande attention. Finalement, une étude expérimentale et/ou de modélisation complète est menée afin de faire progresser la technologie et de justifier ses applications potentielles. Lorsque ce transfert de technologie est susceptible d'avoir lieu (durant les étapes de définition ou de faisabilité du développement produit), il s'accompagne d'un effort important sur la conception produit. Seider [SEI 10] ont adapté cette démarche pour le génie des procédés et l'ont déclinée pour chaque type de produits chimiques.

2.3 Problématique

Les travaux de recherche présentés dans ce mémoire s'inscrivent dans le domaine du « Process Systems Engineering (PSE) » du génie des procédés, à savoir le développement de modèles, méthodes et outils pour l'aide à la conception en génie des procédés. Or cette problématique a donné lieu à de nombreuses contributions sur le sujet et notamment sur les activités support avec des méthodes et outils d'aide à la conception et d'aide à la décision. Cependant, face à l'évolution du génie des procédés et à la complexité grandissante de la conception des procédés et des opérations unitaires, il faut développer de nouvelles méthodes et de nouveaux outils, complémentaires de ceux existants, pour venir améliorer l'efficacité de cette activité support. De manière générale, la démarche présentée dans cette thèse reste proche des contributions actuelles mais s'en démarque significativement par son positionnement en conception préliminaire.

La conception préliminaire englobe les phases de recherche de concepts et de conception architecturale. La recherche de concepts est une phase de créativité pour générer des concepts de solutions ou solutions de principe répondant aux exigences. La phase de conception architecturale, est une phase durant laquelle sont réalisés des choix technologiques, des matériaux, des composants, de dimensions structurales (pré-dimensionnement). La conception préliminaire est une phase cruciale du processus car les choix sont déterminants pour la suite du projet. La figure 1.4 précise le déclin de l'impact des décisions tout au long du processus de conception [IMT 00]. On y aperçoit une vaste zone d'opportunité dans l'étape de conception préliminaire, cependant très peu d'outils sont disponibles pour aider le concepteur durant cette phase. C'est notamment le cas en génie des procédés.



Source : [IMT 00]

Figure 1.4 Impact des décisions au travers des différentes étapes de conception

D'après [SEU 06] la conception préliminaire est une phase des plus essentielles et des plus décisives puisqu'elle réduit des itérations et élimine les problèmes résultant d'informations insuffisantes ou incorrectes. C'est donc à travers cette activité que les solutions à mettre en œuvre sont déterminées afin de réaliser un système conforme aux exigences. De nombreuses décisions sont arrêtées durant cette activité. Remettre en cause l'un des choix génère un allongement de l'activité. Il est donc primordial de réaliser des choix judicieux pour éviter les itérations et pour converger plus rapidement vers une solution pertinente répondant à la majorité des exigences. Or l'exploitation de l'expérience et les connaissances acquises lors de conceptions antérieures permet de diminuer fortement les délais puisque certains choix ne sont plus ni à faire, ni à remettre en cause. A noter également que l'expérience des concepteurs joue un rôle important puisque grâce à elle, il sera possible d'identifier les solutions potentielles, y compris les choix des nouvelles technologies, en fonction des critères économiques et techniques. C'est durant cette étape que la réutilisation d'expériences de conception antérieures peut se révéler intéressante.

Dans ce contexte, il est souhaitable de disposer d'approches et d'outils d'aide à la conception exploitant les connaissances et les savoirs faire. Les entreprises sont conscientes de l'apport concurrentiel que peut générer un système de gestion de connaissances pour l'activité de conception, mais peu d'entre elles se lancent dans la mise en place d'un tel système. La raison principale est qu'il faut un temps considérable pour extraire, formaliser, actualiser et maintenir cette connaissance. De plus un tel projet ne génère pas une valeur ajoutée immédiate et donc devient très coûteuse par rapport à la réalité économique basée sur le court terme. Afin de pallier cet inconvénient, il serait intéressant de capitaliser dynamiquement la connaissance mais peu de recherches ont été menées sur ce sujet.

3 La gestion des connaissances en conception

Pendant la phase de conception, les connaissances jouent un rôle crucial puisqu'elles détermineront la capacité du concepteur à résoudre un problème [AAM 95]. Afin de mieux comprendre l'importance des connaissances dans l'activité de conception, quelques définitions sont présentées par la suite. Elles correspondent au lien : donnée, information et connaissance.

3.1 Définitions

L'élément le plus fondamental de la connaissance est la donnée, celle-ci peut être définie comme suit :

- Une donnée est un élément qui décrit ou qui représente une partie de la réalité, toutefois elle n'a aucun contexte ou interprétation [GRA 00].

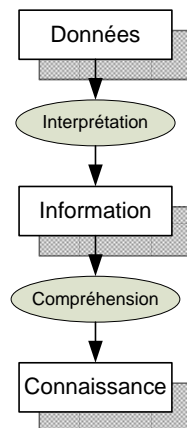
La prise de décisions est basée sur des données, mais celles-ci ne diront jamais ce que l'on doit faire. Elles ne disent rien sur ce qui est important. Malgré tout, les données restent importantes, puisqu'elles sont la base pour la création d'information :

- L'information est, selon [CEN 04] « *le résultat de l'interprétation d'un ensemble de données* »

D'ailleurs le mot « informer » signifie à l'origine « façonner » et sous ce point de vue, l'information est donc capable de former la personne qui la reçoit. L'information, possède un émetteur et un récepteur. L'information est capable de changer la manière dont le récepteur perçoit quelque chose. Elle est capable d'avoir un impact sur ses jugements, ses valeurs et ses comportements. Elle est véhiculée au travers de réseaux formels (qui possèdent une infrastructure visible et définie) et informels (avec une infrastructure invisible). Contrairement aux données, l'information transmet une signification. Non seulement elle informe celui qui la reçoit, mais en plus elle est organisée pour accomplir un certain but. Les données se transforment en information lorsque leur émetteur y ajoutera une signification.

La plupart des gens ont pour sensation intuitive que la connaissance est quelque chose de plus vaste, de plus profond et de plus riche que les données et l'information. Cependant une définition de la connaissance peut paraître imprécise si l'on n'assimile pas la chaîne constituée par la donnée, l'information et la connaissance.

- Selon Davenport et Prusak [D&P 00] la connaissance est le mélange d'expériences, de valeurs, d'informations et d'un « savoir-faire » qui produit un cadre pour l'incorporation de nouvelles expériences, utiles pour l'action.



Source : [A&K 08]

Figure 1.5 La transformation de données en connaissances

Ce qui éclaire immédiatement la définition est que la connaissance n'est pas simple. C'est un mélange de plusieurs éléments, elle est souvent difficile à comprendre d'une façon logique. La connaissance est intrinsèque aux personnes et peut être partagée de plusieurs manières. Par ailleurs, elle peut être vue comme un processus (flux) ou comme un stock. La connaissance dérive de l'information, tout comme l'information est dérivée des données (figure 1.5).

3.2 Les diverses connaissances utilisées en conception

En conception, l'ensemble des connaissances générales et du domaine est la meilleure ressource pour un concepteur chevronné [SCH 83]. La connaissance utile pour la conception peut être définie simplement comme celle qui peut être employée pour produire des solutions [AKE 05]. La connaissance dans le répertoire du concepteur expert est compilée au cours des années, soit par un enseignement conventionnel ou par ses observations. Une fois qu'un concepteur a une tâche spécifique de conception, il commence à rassembler la connaissance spécifique qui pourrait être employée pour accomplir celle-ci.

Une classification des connaissances liées à la conception peut être faite en suivant deux approches. La première, donnant la classification la plus généraliste, distingue deux catégories :

- La connaissance explicite : qui peut être formalisée, structurée et définie d'une manière systématique au travers du langage, de schémas, d'images. Elle est facilement transmise et possède deux dimensions, une individuelle qui représente les connaissances devenues conscientes et une collective qui est traduite sous la forme de règles, normes, codes [COR 06].

- La connaissance tacite : d'après Polanyi [POL 66], la connaissance tacite est celle que l'on possède mais dont on n'est pas capable de l'exprimer. Ainsi la connaissance tacite est un attribut personnel, c'est la connaissance produite par les expériences, valeurs ou l'intuition. Ce type de connaissance est lié à la création de nouvelles connaissances et par conséquent elle est d'une grande valeur pour l'entreprise [COR 06].

La deuxième classification de la connaissance en conception, compte trois catégories possibles :

- La connaissance liée au processus : la connaissance sur les caractéristiques et les propriétés des processus de conception, qui peuvent être employées pour produire un système.
- La connaissance liée au système : elle est basée sur les caractéristiques et les propriétés des systèmes et de leurs matériaux. Elle est appelée « connaissance substantive » [BUN 66].
- La connaissance liée à la réalisation : c'est la connaissance sur les divers processus physiques et chimiques à utiliser pour réaliser un produit, aussi appelée « connaissance opérative » [BUN 66].

Selon Nonaka et Takeuchi [N&T 97], dans chacune de ces trois catégories, un concepteur a la connaissance tacite et la connaissance explicite (aussi appelée codifiée). En ce qui concerne cette dernière on peut faire une autre distinction entre celle basée sur l'expérience de conception développée sur la base d'une abstraction de l'expérience d'un ou plusieurs concepteurs, et celle basée sur l'évidence, établie sur la recherche systématique et formelle.

3.3 Les approches en gestion de connaissances

La gestion des connaissances ou capitalisation des connaissances consiste à collecter le savoir, le mettre en forme, le rassembler dans des bases de connaissances et le rendre disponible [ROL 10]. Les systèmes de gestion de connaissances sont par conséquent des systèmes d'information destinés à gérer le patrimoine de connaissance dans les entreprises. Leurs rôles s'articulent autour de trois actions clé : capitaliser, partager et créer des connaissances. Les objectifs de tels systèmes sont la diffusion des meilleures pratiques, l'amélioration de la prise de décision (meilleures décisions en un temps réduit), la réduction de la subjectivité.

Les entreprises prenant conscience des enjeux et de l'importance de la gestion des connaissances, cette discipline est en plein essor. A l'heure actuelle de nombreuses méthodes et techniques permettent d'identifier, d'analyser, d'organiser, de mémoriser, et de partager des connaissances. Plusieurs de ces approches ont été identifiées dans la bibliographie, et les paragraphes suivants en décrivent quelques-unes.

3.3.1 Approche procédurale

Les systèmes conçus sous ce type d'approche sont basés principalement sur un ensemble de procédures à exécuter, autrement dit un ensemble d'instructions à suivre pour résoudre un problème. Un exemple de cette approche peut être trouvé dans le système Athanor [GUI 06], qui porte sur une approche de gestion de connaissances procédurales pour la maintenance de systèmes complexes.

Les avantages dans les systèmes conçus sur ce type d'approche sont essentiellement leur efficacité pour la résolution de problèmes spécifiques et leur indépendance par rapport au niveau d'abstraction utilisé. Par contre ils s'avèrent inefficaces lorsque le problème à résoudre devient différent des problèmes pour lesquels les procédures ont été conçues.

3.3.2 Approche axiomatique

Cette approche constitue une heuristique permettant d'aider le concepteur à trouver une bonne solution à son problème. Les systèmes sous cette approche consistent à fixer un certain nombre d'axiomes, qui forment les éléments de base de la recherche, et qui permettront d'inférer des conclusions. Une particularité de ces approches est que la qualité de la solution dépend de la créativité, de l'expérience et des connaissances du concepteur. [SUH 90] décrit cette approche comme ayant pour but la définition d'une méthodologie et d'un ensemble de critères rationnels pour la prise de décisions.

3.3.3 Les systèmes experts

« *Un système expert est un ensemble de règles qu'un ordinateur peut suivre en émulant ainsi la prise de décision faite par un expert* » [SCH 99]. Les systèmes experts sont des applications issues de l'informatique qui, sur une base de connaissances, sont capables de résoudre des problèmes demandant de nombreuses connaissances sur un certain sujet. Les

systèmes experts imitent ainsi les activités humaines pour résoudre les problèmes. La connaissance est au cœur de tout système expert.

Pour qu'un système expert devienne un outil effectif, les utilisateurs doivent interagir facilement avec le système. Les systèmes experts nécessitent réunir deux fonctionnalités :

- Ils doivent être conçus en suivant certaines règles compréhensibles de sorte que l'on puisse produire l'explication pour chacune de ces règles, qui se basent sur des faits.
- L'acquisition de nouvelles connaissances : ce sont des mécanismes de raisonnement qui servent à modifier les connaissances précédentes.

Chaque système expert se compose de deux parties principales : la base de connaissances et le raisonnement ou moteur d'inférence [PIG 05]. La base de connaissances contient la connaissance effective et heuristique. La première est la connaissance du domaine qui est largement partagée et que l'on trouve par exemple dans les références bibliographiques. La deuxième est la connaissance moins rigoureuse, plus expérimentale et critique par rapport au fonctionnement. A l'inverse de la connaissance effective, la connaissance heuristique est rarement discutée et est en grande partie individualiste. Son efficacité réside dans la qualité de la connaissance spécifique qu'il contient sur la tâche à accomplir. Pour maintenir les systèmes experts, il faut que les chercheurs continuent à explorer et à ajouter au répertoire actuel, des méthodes de représentation de la connaissance et du raisonnement. Étant donné l'importance de la connaissance dans les systèmes experts, les méthodes d'acquisition de connaissance sont donc d'une importance vitale.

Les systèmes experts présentent un avantage lié à leur efficacité à résoudre des problèmes bien spécifiques. Ils ne nécessitent pas de représentation complète du problème, souvent difficile à obtenir. D'ailleurs, la représentation des connaissances et la génération d'explications sont relativement faciles. A cause de l'incomplétude de la représentation du problème, il est impossible de prouver l'optimalité d'une solution et il est difficile de générer des déductions poussées. Les connaissances ne sont pas transférables à d'autres applications. Il est aussi très difficile de maintenir les connaissances dans de tels systèmes.

3.3.4 Les approches à partir de cas

De façon générale, les systèmes à base de connaissances permettent le stockage et la consultation de connaissance, le raisonnement automatique sur les connaissances stockées, leur modification ainsi que le partage de connaissances entre systèmes informatiques [BAH

06]. Parmi ces systèmes on trouve le Raisonnement à Partir de Cas (RàPC) qui doit son développement aux travaux issus des sciences cognitives et plus particulièrement de ceux de Schank [SCH 82] sur la mémoire dynamique. Le RàPC a pour but de résoudre un nouveau problème à l'aide d'une base de cas [REN 07A]. Dans cette approche, le cas est la notion centrale qui représente une expérience passée. Un cas se décompose donc en deux parties : la description du problème et la description de la solution. De nombreux cas sont ensuite regroupés dans une mémoire appelée base de cas. Ainsi face à un nouveau problème le système de RàPC est capable de remémorer de sa base de cas, un (ou plusieurs) cas similaire dont la solution peut être proposée comme une nouvelle solution pertinente à la requête.

Le RàPC ne nécessite pas une connaissance explicite du problème. Il suffit d'en identifier les principales caractéristiques. Il permet l'apprentissage du système par l'introduction de nouveaux cas résolus. Une explication est facile à construire : c'est l'ensemble des cas similaires retenus dans la résolution. Au début, la base de connaissances peut être alimentée facilement grâce à des cas réels connus. Néanmoins quelques inconvénients se posent lors de son implantation, comme par exemple : une limitation de l'utilisation à des problèmes voisins, l'absence de garantie d'optimalité des solutions et la nécessité de disposer d'un grand espace mémoire pour stocker tous les cas rencontrés. Ces aspects seront traités plus en détail dans le chapitre 2.

3.3.5 Les approches à base de contraintes

La programmation par contraintes (PPC) a maintenant fait ses preuves pour la résolution de problèmes combinatoires complexes dans le domaine de l'aide à la décision. La PPC est basée sur des travaux dans divers domaines comme les mathématiques discrètes, l'analyse numérique, et la recherche opérationnelle.

La connaissance en PPC est modélisée sous forme de contraintes : relations logiques, expressions mathématiques, domaines de validité. Le problème est soumis au modèle de connaissance via un certain nombre de variables. Une résolution est alors conduite à travers les contraintes pour réduire le domaine de validité des variables. En conception, la modélisation des connaissances constitue la principale difficulté pratique de la PPC, car une fois représenté dans le formalisme voulu, un problème peut être résolu automatiquement à l'aide d'un solveur adapté. Le solveur déterminera si le problème est consistant et, le cas échéant, fournira une solution.

L'inconvénient majeur de cette approche est qu'elle nécessite un effort conséquent pour identifier et formaliser les connaissances et bâtir le modèle de raisonnement. L'évaluation du temps d'exécution d'un programme de PPC peut être dépendante du problème et il n'est pas possible d'énoncer des règles claires concernant les performances temporelles, ce qui s'accommode mal avec les exigences de certains projets. Mais, les systèmes à base de contraintes ont l'avantage de pouvoir fournir des solutions inédites ou d'établir qu'un problème n'a pas de solution. Ils permettent également de considérer les aspects fonctionnels et techniques simultanément. Ils visent à remédier aux inconvénients des méthodes de résolution actuellement utilisées et à proposer au final un outil de résolution offrant une bonne flexibilité et permettant de proposer des solutions qui satisfont toutes les contraintes du problème. Cette approche est détaillée dans le chapitre 3.

3.3.6 Les approches à base de modèles

Dans un problème de conception, cette approche offre une vue globale du problème, la possibilité d'aborder le problème de différents points de vue, la garantie d'optimalité des solutions, l'objectivité du raisonnement et la génération d'explications plus précises. Elle permet de simuler et d'expliquer le comportement de l'objet. L'approche permet aussi une réutilisation relativement facile des composants du modèle pour d'autres problèmes.

La difficulté de cette approche se situe au niveau de l'obtention d'un modèle valide adapté au problème. La validation d'un modèle nécessite des efforts considérables. Au niveau de leur utilisation, elle peut être relativement gourmande en calcul et nécessite un certain niveau de connaissances pour sa manipulation et sa maintenance (voir point 4.1).

3.3.7 L'intelligence collective

Cette approche porte sur l'étude de la structure des interactions se déroulant au sein d'un groupe, afin de proposer des outils et des méthodes de structuration permettant une « *meilleure mise en valeur des savoirs échangés et garantissant une réutilisation plus aisée* » [Z&G 01]. Dans ce type d'approche, on note les travaux de Darses [DAR 05] qui utilise les résultats du domaine de l'ergonomie cognitive pour construire un modèle d'expertises des activités basé sur la coopération afin d'identifier les connaissances relatives à la démarche de résolution de problèmes.

Le modèle DIPA de Lewkowics [L&Z 99] représente la logique de conception sous forme d'une analyse cognitive d'une résolution de problème. Ce formalisme (données, interprétations, propositions, accord) utilise des méthodes de résolution de problèmes définies dans l'ingénierie des connaissances pour structurer une prise de décision. Il vise à structurer et garder les traces de résolution de problèmes en conception. Cet aspect souligne l'importance du rôle du partage de connaissances au sein de communautés, car la connaissance résultante est bien plus riche que les connaissances individuelles.

3.4 La gestion des connaissances en génie des procédés

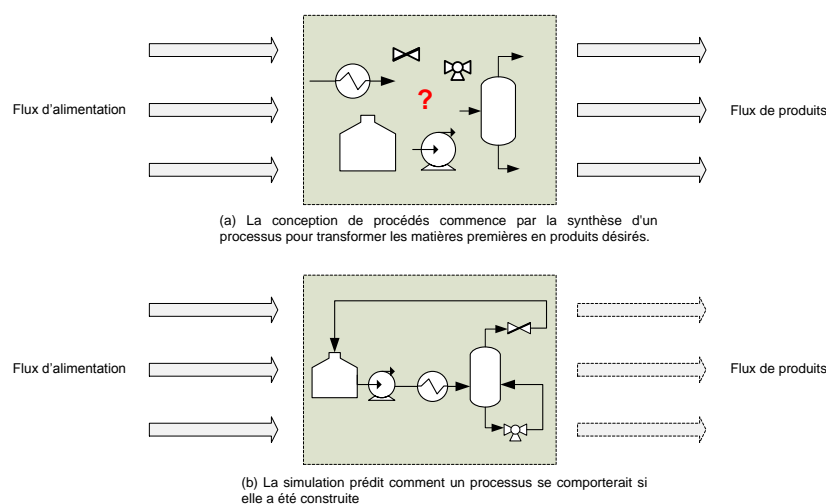
Traditionnellement, la gestion des connaissances en génie des procédés se limitent aux connaissances explicites sous forme de modèles mathématiques. Ces modèles permettent de les capitaliser grâce à l'écriture d'équations, de corrélations, de courbes. Ils traduisent les phénomènes physico-chimiques mis en œuvre dans un procédé. Toutefois, avec la modélisation-simulation seules les connaissances explicites sont capitalisées ce qui limite la portée de cette approche. A cause des évolutions de la discipline abordée dans ce chapitre et de la plus-value de la connaissance tacite, il devient primordial de proposer de nouvelles approches pour essayer de capitaliser et diffuser les savoirs-faire.

La recherche actuelle [Z&D 06] démontre qu'il y a toujours un besoin en outils qui capitalisent, diffusent et génèrent de la connaissance de conception mais également qui facilitent la prise de décision de l'expert mais aussi son intervention durant la résolution. Cette problématique est abordée par la mise en œuvre et l'adaptation d'approches issues du Génie Industriel et plus particulièrement de l'Intelligence Artificielle au génie des procédés afin d'instrumentaliser la conception préliminaire.

4 Le rôle de la modélisation-simulation en génie de procédés

Dans le domaine des procédés, les concepteurs représentent les transformations individuelles (opérations unitaires) et leurs connections grâce à des modèles qui permettent une meilleure analyse du procédé. Le « flowsheet » est par exemple l'un de ces modèles qui permet de représenter tout le procédé à travers des diagrammes qui symbolisent les opérations unitaires et leurs connections. Une fois la structure de ce flowsheet définie, une simulation du procédé peut être conduite [SMI 05].

La simulation peut être définie comme « *une représentation d'un système réel pour effectuer des expériences, et comprendre le comportement de ce système ou d'évaluer plusieurs stratégies (dans les limites imposées par un critère ou par un ensemble de critères) pour sa mise en œuvre [RIO 00]* ». D'après cette définition, on constate que la modélisation-simulation cherche à imiter le comportement d'un système réel. Ce modèle permet d'analyser son fonctionnement et d'évaluer les effets des phénomènes qui y interviennent. La figure 1.6 présente les deux activités principales de la modélisation-simulation d'un procédé, dont la première étape regroupe les transformations individuelles, et la seconde les interactions permettant ainsi une simulation.



Source : [SMI 05]

Figure 1.6 La modélisation-simulation d'un procédé

En PSE on cherche à ce que les modèles conçus puissent être réutilisés pour d'autres systèmes similaires ou qu'ils soient intégrés dans un simulateur des procédés. D'où l'importance de l'habilité des ingénieurs PSE lors de la modélisation et la résolution d'un système.

4.1 Le modèle mathématique et la capitalisation des connaissances en génie des procédés

En général, un modèle peut être décrit comme « *un ensemble d'informations relatives à un système qui permet son étude* » [FIS 78]. Un modèle peut également être défini comme la représentation d'une théorie ou d'une observation empirique grâce à laquelle on peut manier un problème et trouver, au moyen de différentes techniques, la meilleure solution à ce problème. De ce constat on remarque que, plus un modèle est proche du système qu'il représente, plus il aura des chances pour le résoudre, avec un haut degré de précision.

En génie des procédés le terme modèle fait référence à un ensemble d'équations mathématiques pour tenter de représenter le plus fidèlement possible le comportement d'un système réel afin de l'étudier. La modélisation cherche à traduire une réalité souvent complexe à cause d'un ensemble de phénomènes combinés, au travers d'un modèle ayant un comportement semblable. Ce modèle saisit rarement la réalité dans toute sa complexité mais se limite à un certain point de vue de celle-ci en fonction des connaissances et des informations disponibles, du contexte d'utilisation et de l'objectif visé. Ainsi différents modèles avec des hypothèses plus ou moins restrictives sont développés durant les différentes situations de vie d'un procédé :

- En conception préliminaire : pour l'étude de faisabilité, le développement et le pré-dimensionnement d'alternatives de conception afin d'alimenter les critères de choix pour en réduire le nombre.
- En conception détaillée : pour le dimensionnement, la sélection des conditions opératoires, l'optimisation et le contrôle du procédé, voire la reconfiguration.
- En phase d'exploitation : pour piloter l'installation, en améliorer le fonctionnement, proposer de nouvelles modifications du procédé existant, mener des études de sécurité ou d'impact environnemental...

Ces situations de vie mettent en évidence les deux principales cibles de la modélisation : la supervision des procédés d'une part, l'acquisition et la capitalisation de connaissances d'autre part. A noter que pour un système donné, il n'existe pas un modèle unique de représentation mais plusieurs possibilités plus ou moins détaillées en fonction de la finalité de l'étude.

Concernant l'acquisition et la capitalisation des connaissances, le modèle aide à comprendre les phénomènes impliqués dans le système et d'en optimiser le fonctionnement en fonction du critère poursuivi. Dans sa version finale, le modèle sert de capitalisation d'une grande partie des connaissances acquises sur le système. En effet, il va traduire la connaissance théorique que le modélisateur a du système et de l'information expérimentale disponible. Ainsi, il est à noter la complémentarité entre modélisation et étude expérimentale. Même si la phase de conception est largement dominée par la modélisation simulation afin de réduire le temps et les coûts de développement, le passage sur pilote est une étape importante et primordiale pour la validation du modèle mais également pour l'acquisition de connaissances complémentaires (paramètres physiques, phénomènes mis en jeu...) afin d'affiner le modèle avant le passage à l'échelle industrielle. Modélisation et études

expérimentales sont donc très liées pour conduire à un approfondissement itératif des connaissances sur le système.

4.1.1 Développement d'un modèle de procédés

Généralement un modèle en conception doit comporter un ensemble de caractéristiques qui le rendent plus performant par rapport à ses objectifs. Les parties qui caractérisent donc un modèle sont [ROL 07] :

- **Les composants** sont les parties constitutives du problème à modéliser. On les appelle aussi des éléments ou des sous-systèmes.
- **Les variables** sont des valeurs qui peuvent changer et font partie des contraintes du modèle ou d'une fonction objectif.
- **Les constantes** sont des paramètres auxquels on assigne des valeurs bien définies. Une fois ces paramètres établis, ils restent constants tout au long du problème.
- **Les relations fonctionnelles** démontrent le comportement des variables et les paramètres d'un composant du système.
- **Les contraintes** sont des limitations imposées aux valeurs des variables où la manière avec laquelle les ressources doivent être attribuées.
- Dans **les fonctions objectifs**, on définit explicitement les objectifs du système et la façon dont ils seront évalués.

Durant le processus de conception, il existe aussi certaines caractéristiques souhaitables pour qu'un modèle puisse décrire rigoureusement le système tout en produisant des réponses rapides et fiables :

- Être complet en prenant en compte la multitude d'alternatives présentes dans la phase initiale du processus.
- Présenter un haut degré d'adaptabilité. En effet, un modèle doit pouvoir s'adapter à plusieurs problèmes. L'idée de base étant celle de ne pas être ni trop général ni trop restrictif.
- Être simple et robuste. Il convient de commencer avec des modèles simples et robustes qui permettent de produire rapidement des informations pour réaliser les premiers choix de conception. Par la suite, il faudra ajouter le degré de complexité nécessaire en fonction de l'objectif recherché.
- Faisable au niveau des ressources disponibles, choix technologiques...

- Economique : le coût maximal investi dans le développement d'un modèle doit être au moins égal au bénéfice minimum que l'on obtient.

La conception d'un modèle et la réalisation d'expériences avec ce dernier, ont pour but de comprendre le comportement d'un système ou d'évaluer des stratégies à mettre en place. Ceci implique d'établir certains critères que doit respecter tout modèle :

- Compréhensible de la part du concepteur.
- Capable de fournir des solutions cohérentes aux objectifs.
- Facile à contrôler et à manipuler par l'utilisateur.
- Evolutif et générique, c'est-à-dire, qu'il ait la capacité d'être extrapolé à des problèmes beaucoup plus complexes selon les exigences du concepteur.

Bien qu'elles soient génériques, toutes les caractéristiques et remarques antérieures peuvent aussi être prises en considération lors du développement d'un modèle en génie des procédés. D'ailleurs pour la construction de modèles dans ce domaine [MEY 04] a formalisé l'approche communément utilisée en PSE sous la forme d'une démarche procédurale avec les étapes suivantes :

1. Identification de l'objectif principal de la modélisation.
2. Caractérisation du système matériel étudié : comportement thermodynamique, cinétique...
3. Identification des principaux phénomènes dont ceux qui sont limitants, à partir d'expériences ou de connaissances auprès d'experts.
4. Sélection d'une base théorique : énoncé des hypothèses du modèle.
5. Formulation des équations.
6. Analyse des degrés de liberté.
7. Choix de la méthode de résolution.
8. Estimation des paramètres.
9. Validation du modèle sur des données expérimentales.
10. Documentation du modèle.
11. Intégration du modèle dans un environnement de simulation (impact de la nouvelle unité sur l'ensemble du procédé).
12. Développements ultérieurs du modèle.

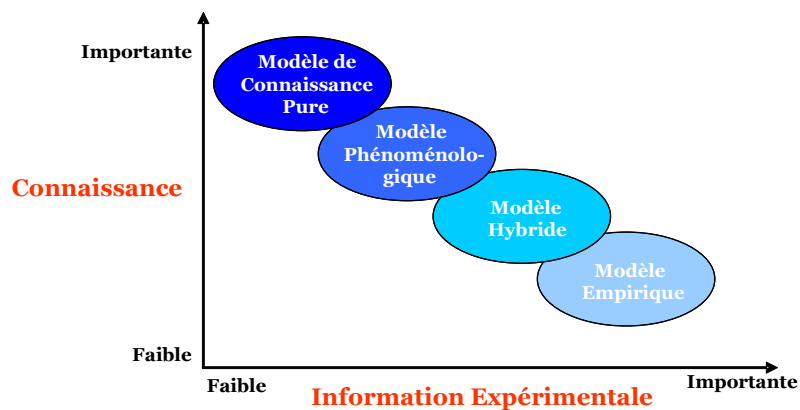
Cette démarche a l'avantage de clairement démontrer que la modélisation des procédés est une activité qui demande des connaissances diverses et variées. En effet développer un

modèle ne se résume pas à la simple écriture des équations mais résulte d'une collaboration multidisciplinaire entre les connaissances acquises sur le système étudié, et les connaissances mathématiques sur l'écriture, la formulation et la résolution des équations qui doivent conduire à une solution physique. L'expertise du physicien est primordiale pour les étapes 3, 4, 5 et 9. Il aide à construire l'abstraction de la réalité grâce à sa contribution cruciale lors de la formulation des hypothèses simplificatrices les plus réalistes. Ces hypothèses découlent directement de la connaissance des phénomènes importants. Il est également d'une aide précieuse pour la validation expérimentale et la modification du modèle en fonction des connaissances nouvelles acquises (à noter que l'inverse est également vraie, le modèle permet de trouver des conditions expérimentales propices à l'amélioration des performances). Les étapes 5, 6 et 7 relèvent des mathématiques appliquées et de l'informatique. Quant aux étapes 1, 5 et 8 elles demandent la mise en place d'une équipe multidisciplinaire.

4.1.2 Les différents types de modèle en procédés

Il existe différents types de modèles que [MEY 04] a classés en quatre grandes catégories en fonction des connaissances théoriques et de l'information expérimentale disponible (figure 1.7).

Les modèles de connaissance pure sont basés uniquement sur la connaissance théorique du système. Ils demandent une connaissance approfondie sur le système étudié et notamment sur les phénomènes. Ils cherchent à modéliser ces derniers à l'aide des lois fondamentales de la physique et de la mécanique. Ce travail approfondi limite le besoin en information expérimentale (nombre de paramètres à identifier limités) mais en contrepartie le temps de développement et la difficulté de simulation sont accrus. Bien que ces modèles offrent une meilleure cohérence, extrapolabilité, un plus vaste domaine d'application, ils reposent fortement sur la compréhension du système. Une erreur d'appréciation peut entraîner un modèle erroné avec des conséquences importantes lors de son utilisation. Toutefois, parmi les différents types de modèle, c'est celui qui procure la meilleure façon d'acquérir et de capitaliser la connaissance de par l'analyse et la réflexion importante mais néanmoins nécessaires sur le système.



Source : [MEY 04]

Figure 1.7 Les différents types de modèle et leurs niveaux de connaissance

Quant aux modèles phénoménologiques, ils mêlent des lois phénoménologiques (hydrodynamique et transferts) et des corrélations empiriques. Dans la mesure du possible, ils s'appuient au maximum sur les lois de la physique mais certains phénomènes sont encore représentés par un modèle comportemental par souci de simplification de la résolution du modèle ou par manque de connaissance. La substitution de la loi physique correspondante est réalisée soit par une corrélation empirique soit par des hypothèses de fonctionnement. C'est souvent le cas pour l'aspect hydrodynamique des unités (écoulement piston, phase parfaitement agitée...) même si grâce à la mécanique des fluides numériques des analyses plus fines permettent de lever ces hypothèses. La raison d'être de ces modèles est que les modèles de connaissance pure sur des systèmes complexes restent encore un véritable défi.

Les modèles hybrides résultent du couplage entre bilans matière et énergie et des réseaux de neurones (par exemple pour modéliser le comportement de systèmes réactifs). Ils sont plus particulièrement appliqués pour la modélisation de systèmes mettant en œuvre des réactions complexes comme par exemple dans les réacteurs biologiques. Le modèle de fonctionnement du système est basé sur les bilans matière et énergie, les réseaux de neurones servant à modéliser la partie réactive.

Enfin les modèles empiriques sont basés uniquement sur l'information expérimentale. Ils sont construits dans le but de forcer une structure mathématique à obéir à des observations expérimentales comme par exemple : réponse à un échelon ou impulsion, analyse statistique multi variée, fonctions de transfert... La rapidité de développement, la facilité de simulation sont à l'origine de leurs utilisations pour le contrôle de systèmes non linéaires (même lorsque les phénomènes mis en jeu sont bien connus). En revanche, ces

modèles s'appuient sur des données expérimentales en quantité. De plus, ils sont peu extrapolables en dehors du domaine de réalisation de l'étude.

4.1.3 L'évaluation d'un modèle

Dans la littérature il est rare de trouver un moyen d'évaluer la performance d'un modèle, toutefois les travaux de Vernat [VER 04] exposent une approche intéressante pour évaluer l'adéquation d'un modèle par rapport aux performances attendues. Cette approche propose l'utilisation de quatre critères d'appréciation intrinsèques à un modèle. Ses critères d'évaluation correspondent à deux critères associés à son exploitation (parcimonie, spécialisation), et deux autres associés aux résultats (exactitude, précision) :

- la parcimonie représente la capacité d'un modèle à décrire correctement le comportement du système avec un nombre minimum de paramètres et d'équations.
- la spécialisation d'un modèle sert à désigner le domaine d'application auquel il est restreint.
- l'exactitude, est définie par [TRA 97] comme « *la distance entre la solution du modèle et le comportement de référence, qui est généralement le comportement observé du système réel* ».
- la précision définit l'étendue du domaine des valeurs possibles d'une variable. Elle est due soit au niveau de représentation du modèle, soit à l'incertitude liée aux variables.

D'après Vernat [VER 04] « *les niveaux de parcimonie et de spécialisation du modèle vont conditionner les possibilités d'utilisation directe du modèle (temps de calcul, outil de traitement du modèle utilisable ou non)* » et « *l'exactitude et la précision du modèle vont conditionner la validité de l'utilisation du modèle* ». Selon ces quatre critères un modèle est meilleur d'un point de vue de l'aide à la décision lorsqu'il est parcimonieux, exact, précis et peu spécialisé.

4.2 Les avantages et limites de la modélisation

Bien que l'approche par modèle offre (suivant le niveau d'abstraction du modèle utilisé) une vue globale du problème, la possibilité d'aborder le problème de différents points de vue, l'objectivité du raisonnement et la génération d'explications précises (élément important car les connaissances expliquées sont mieux comprises et préférentiellement réutilisées), elle présente quelques difficultés et limites parmi lesquelles on distingue notamment [ROL 07] :

- La modélisation offre des valeurs estimées sur les éléments du problème, qui pourraient parfois ne pas correspondre aux valeurs réelles du système.
- Concevoir un modèle peut demander un temps important.
- Les techniques de modélisation-simulation nécessitent l'utilisation de ressources (e.g. logiciels, ordinateurs puissants) qui peuvent représenter un investissement important.
- Au fur et à mesure qu'augmente la complexité ou le nombre des variables du modèle, augmente la difficulté et le temps de calcul.

5 Conclusion

Dans ce premier chapitre une synthèse des différentes problématiques liées à la conception de produits et procédés a été présentée. Il positionne la conception comme une étape cruciale pour relever les défis techniques, économiques, environnementaux et sociaux des entreprises. La phase de conception en Génie des Procédés, est un processus complexe demandant des collaborations entre diverses disciplines scientifiques ce qui a pour conséquence d'accroître l'intégration et l'utilisation de connaissances de multiple domaines dans le but de maîtriser les procédés dans leur totalité.

Ce chapitre démontre également qu'un modèle peut être un moyen de capitaliser et augmenter la connaissance sur un système. Pour construire un modèle, il faut comprendre et traduire mathématiquement les causes et conséquences des effets de chaque phénomène. De plus, il faut choisir une bonne formulation parmi un ensemble d'alternatives mais aussi choisir la méthode numérique la mieux adaptée à la nature et aux caractéristiques du système d'équations. On remarque que les modèles restent souvent à un certain niveau de généralité, ne prenant pas toujours en compte l'expertise technique sur les choix technologiques ou la configuration technique des systèmes. Toute cette connaissance amont de l'expert technique ayant guidée ces choix de conception n'est pas capitalisée : type d'agitateur, choix des matériaux, de géométrie de l'opération unitaire, type et taille d'un garnissage... Bien que la modélisation simulation offre une voie pour la capitalisation des connaissances, cette dernière n'est que partielle. Le modèle ne peut pas à lui seul capitaliser l'ensemble des connaissances nécessaire à la conception d'un système.

Dans ce cadre, un modèle de conception préliminaire est proposé, inspiré des approches de conception présentées dans ce chapitre. Il cherche à mettre en valeur l'expertise du concepteur avec des connaissances disponibles sur le domaine afin de les exploiter et

construire un modèle qui représente le problème. Celui-ci est soumis à une résolution pour obtenir une solution qui est révisée par le concepteur. Après validation la solution peut être utilisée pour la phase de conception détaillée. A noter que la nouvelle expérience peut être stockée dans une base de connaissances pour une future utilisation. Le modèle est représenté par la figure 1.8.

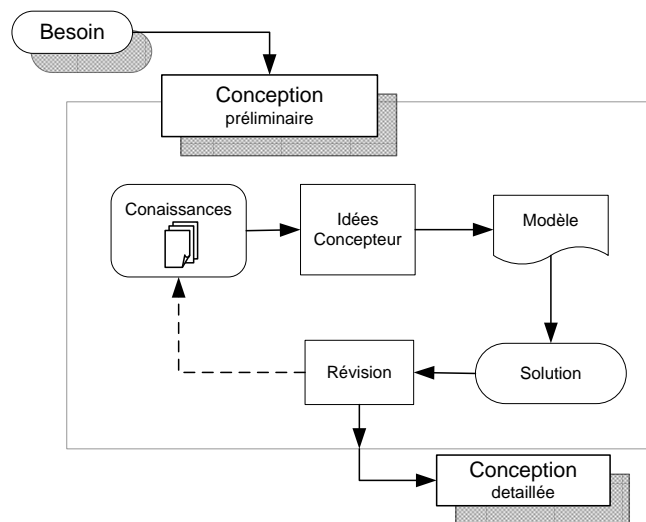


Figure 1.8 Le modèle de conception préliminaire

Parallèlement, la mise en place de nouvelles méthodes et de nouveaux outils est une des voies envisageables pour pallier les problèmes rencontrés en conception. Toutefois, face à la création de nouveaux produits ou procédés, les connaissances à mettre en place peuvent se raréfier voire être inexistantes. Les connaissances doivent alors être extraites ou créées par des méthodes beaucoup plus performantes. C'est dans ce cadre que la gestion de connaissances a été proposée, afin d'acquérir et de capitaliser les connaissances capitales pour la conception en Génie des Procédés. Parmi les différentes approches présentées (basées sur de techniques d'Intelligence Artificielle), le RàPC et la PPC semblent être les plus adaptées pour soutenir la conception mais aussi pour proposer de l'innovation incrémentale⁴. Par la suite, ces techniques seront expliquées plus en détail afin de poser les bases d'une approche envisageant leur couplage.

⁴ Elle relève de l'amélioration continue de produits ou procédés dans le cadre de trajectoires d'évolution déjà définies.

Le Raisonnement à Partir de Cas

Le Raisonnement à Partir de Cas est une méthode qui permet de résoudre des problèmes en s'appuyant sur un principe général : des problèmes similaires ont des solutions similaires. Ainsi, la notion centrale est un cas qui représente une expérience passée contextualisée. De nombreux cas sont ensuite regroupés dans une mémoire de cas. Lorsqu'un nouveau problème est rencontré, les différentes étapes de la méthode permettent d'extraire un (ou plusieurs) cas similaire pertinent dont la solution sera utilisée pour proposer une solution adéquate au problème courant. Cette approche de résolution de problèmes est décrite dans ce chapitre ainsi que ses applications dans le domaine de la conception en génie des procédés.

Chapitre 2 : Le Raisonnement à Partir de Cas

1 Introduction

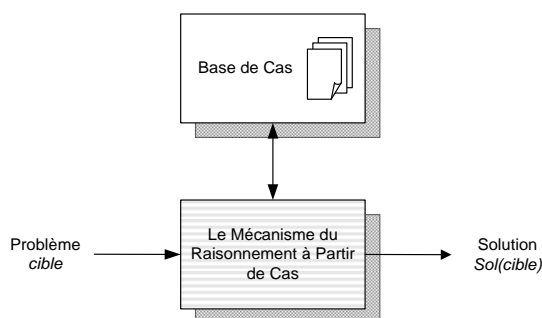
Le Raisonnement à Partir de Cas (RàPC) trouve son origine et son développement dans les travaux issus des sciences cognitives et plus particulièrement de ceux de Schank [SCH 82] sur la mémoire dynamique. Le principe général du RàPC est : des problèmes similaires ont des solutions similaires. Ce mode de raisonnement pour la résolution de problèmes s'appuie sur une base d'expériences passées pour résoudre de nouveaux problèmes. A l'heure actuelle le RàPC évolue grâce à des recherches dans différentes disciplines comme l'intelligence artificielle, les machines apprenantes et les mathématiques [R&A 05]. Le travail de [FUC 97] et l'ouvrage de [P&S 04] fournissent une introduction et une bonne synthèse sur le RàPC.

1.1 Éléments fondamentaux

Le cas est la notion centrale du RàPC, il décrit une expérience passée contextualisée représentant un épisode de résolution de problème. Un cas se compose donc de la description d'un problème (élément de l'espace des Problèmes) et de sa solution (un élément de l'espace des Solutions) associée (relation binaire entre les deux espaces) et éventuellement d'une troisième partie pour des informations complémentaires. Une base de cas regroupe un ensemble fini de cas, elle est généralement munie d'une structure (figure 2.1). Résoudre un nouveau problème (appelé problème cible et noté *cible*) avec un système de RàPC, revient à trouver ou à construire sa solution (notée $Sol(cible)$) en s'appuyant sur les cas stockés dans la base, appelés cas sources (notés (*source*, $Sol(source)$)). Grâce à son traitement par un raisonnement, un cas constitue une unité de connaissance et non pas une simple donnée. De par leur apparente simplicité conceptuelle et leur efficacité, les systèmes de RàPC sont utilisés dans de nombreux domaines, ce qui permet de les classer en fonction du type d'application : aide à la décision, diagnostic, configuration, planification, etc.

Un exemple pour mieux comprendre le fonctionnement d'un système de RàPC, peut être observé lors de la création d'un procédé. Dans cette situation, un cas peut représenter des différents paramètres d'une conception passée, spécifications, conditions opératoires, modèles thermodynamiques, diagrammes... Face à la nécessité de création d'un nouveau procédé, l'ingénieur peut comparer ses besoins à ceux des conceptions passées résumées par exemple sous la forme d'un flowsheet. Un flowsheet similaire peut alors être utilisé et

modifié, si nécessaire, pour être adapté à la nouvelle situation. Cet exemple met en évidence qu'un système de RàPC est capable de résoudre de nouveaux problèmes en adaptant les solutions de problèmes passés. Par conséquent, le RàPC devient un système capable de « raisonner » en s'appuyant sur un cas semblable aux spécifications de la requête.



Source : adaptée de [P&S 04]

Figure 1.1 Le Système du RàPC
Figure 2.1 Principe de fonctionnement du Raisonnement à Partir de Cas

1.2 Le cycle du RàPC

Le RàPC peut être défini comme une méthodologie capable d'effectuer du raisonnement et de l'apprentissage, se basant sur les connaissances spécifiques des problèmes passés [KOL 92]. Il réutilise donc cette connaissance pour fournir une solution à un nouveau problème. Différents modèles furent proposés pour représenter les différentes étapes séquentielles du processus déployé dans un RàPC : [ALL 94], [HUN 95] et [LEA 96]. Actuellement le cycle R^5 proposé par [F&S 03] est communément utilisé pour modéliser et construire un système de RàPC (figure 2.2). Ce cycle est une extension du modèle R^4 introduit par [A&P 94]. Dans le cycle R^5 chaque R fait référence à une des étapes suivantes :

- **Représenter** : Il s'agit de mettre en forme le problème *cible* 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 la plus communément utilisée sous forme d'un vecteur d'attributs valeurs est retenue par la suite.
- **Remémorer** : Son objectif est de sélectionner et d'extraire un cas *source* similaire à *cible*. Le point clef de cette étape est la mesure de similarité.

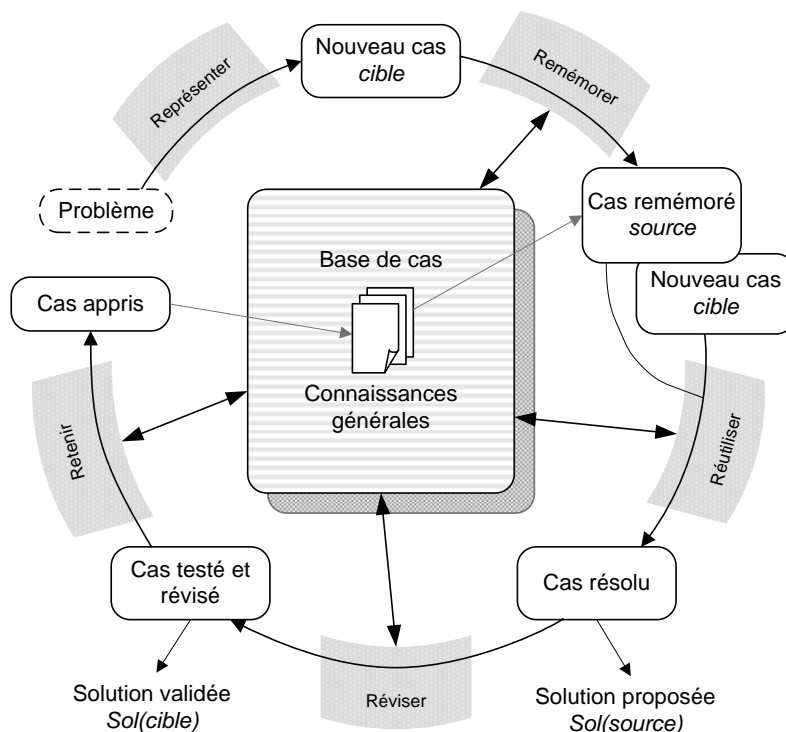
$(cible, Base\ de\ Cas) \rightarrow (source, Sol(source)) \in Base\ de\ Cas$

- **Réutiliser** : En s'appuyant sur le cas remémoré, la réutilisation cherche à résoudre *cible*. Le système RàPC adapte la solution d'un cas remémoré pour répondre aux exigences du nouveau problème. Cette phase est aussi connue sous l'appellation « l'adaptation de cas », elle peut être formalisée par :

$(source, Sol(source), cible) \rightarrow Sol(cible).$

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 pour $cible$, sans aucun changement (utilisé pour l'aide à la décision ou justifier un choix). Mais la plupart du temps, cette étape d'adaptation cherche à définir l'écart entre les problèmes $source$ et $cible$ pour ensuite modifier $Sol(source)$. Les méthodes d'adaptation doivent répondre aux deux questions suivantes : Qu'est ce qui devrait être modifié dans $Sol(source)$? Comment ce changement doit-il être opéré ?

- *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 d'échec du test, 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* : S'il est opportun, apport d'une réelle valeur ajoutée à la base de cas, le nouvel épisode de résolution 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 dans un tel système.



Source : adaptée de [A&P 94] et [F&S 03]

Figure 2.2 Le cycle R^5 du Raisonnement à Partir de Cas

En réalité, le modèle R⁵ est bien plus complexe que cette présentation sommaire car chaque étape englobe de nombreux sous processus spécifiques dernièrement détaillés par [LOP 06]. [FUC 97] en donne également une première introduction complète.

Selon [P&S 04] dans de nombreuses applications pratiques, il est parfois difficile de distinguer les étapes de réutilisation et de révision, et plusieurs chercheurs les concatènent en une étape unique : l'adaptation. Cependant, l'adaptation dans les systèmes de RàPC est toujours une question ouverte car cette dernière englobe un processus compliqué qui essaye de manipuler les solutions des cas. Habituellement, cela exige le développement d'un modèle causal entre l'espace des problèmes (i.e. les spécifications) et l'espace des solutions (i.e. les caractéristiques des solutions).

1.3 Les connaissances dans le RàPC

D'après Cordier [CFM 06] «...en RàPC, les connaissances de base sont les expériences de résolution de problème : les cas... ». Tout système de raisonnement à partir de cas effectue des raisonnements en s'appuyant sur les connaissances qu'il exploite. Les cas étant la source principale, mais d'autres systèmes utilisent différentes sources de connaissances, on distingue notamment les « connaissances du domaine ». Plus ces connaissances du domaine sont exactes et précises, meilleures seront les inférences effectuées par le système de RàPC [COR 07]. [CFM 06] signale que les systèmes de RàPC sont des systèmes à base de connaissances qui exploitent quatre conteneurs de connaissances distincts à savoir : le vocabulaire de description du domaine, les cas, les connaissances de similarité et les connaissances de transformation de la solution aussi appelées connaissances d'adaptation, tableau 2.1.

Les systèmes de RàPC ont longtemps été considérés comme des alternatives intéressantes aux systèmes à base de règles car ils demandent théoriquement un moindre effort d'ingénierie des connaissances pour être mis en œuvre. [CFM 06] soutiennent que « *le RàPC est particulièrement adapté aux situations dans lesquelles la théorie du domaine est faible ou peu formalisable. Il a longtemps été argumenté que le RàPC était une solution au goulet d'étranglement de l'acquisition des connaissances dans la mesure où il est plus facile de collecter un ensemble de cas que de construire une base de connaissances* ». En revanche, le RàPC ne permet pas de se passer d'une base de connaissances et on se trouve confronté au problème de leur acquisition avec notamment des ontologies du domaine, les connaissances de similarité ou les connaissances d'adaptation [CFM 06].

Type de connaissance	Forme	Étape d'acquisition	Méthode
Cas	Parties problème et solution (ensembles de descripteurs), Traces du raisonnement (étapes conduisant du problème à la solution)	Conception : utilisation de cas connus pour "entraîner" le système Mémorisation : mise en mémoire du cas résolu lors d'un cycle de raisonnement	Classification Indexation
Connaissances du domaine	Concepts : propriétés et relations avec d'autres concepts, Règles, Dépendances	Acquisition initiale relativement aisée si la théorie du domaine est faible	Description et modélisation par l'expert
Connaissances de similarité	Mesures numériques prédéfinies, Mesures empiriques basées sur la comparaison de descripteurs, Mesures plus complexes avec prise en compte de l'adaptabilité, Poids, Chemins de similarités, Etc	Acquisition initiale non aisée, pas de méthodologie pour les concevoir, Mémorisation : acquisition de nouvelles connaissances et amélioration des connaissances existantes	Modélisation par l'expert, Apprentissage introspectif, Apprentissage symbolique automatique (fouille de données, réseaux de neurones ...) Etc
Connaissances d'adaptation	Règles d'adaptation, Opérateurs d'adaptation, Cas d'adaptation		

Source : [CFM 06]

Tableau 2.1 La typologie des connaissances du RàPC

Les connaissances de similarité sont utilisées pour remémorer les cas candidats et les connaissances d'adaptation pour adapter les solutions des cas remémorés. Or, l'expérience montre que les connaissances de similarité et d'adaptation disponibles au départ sont difficiles à modéliser car elles sont imprécises ou incomplètes, et de plus, elles peuvent évoluer au cours du temps. Par conséquent, il est souhaitable de disposer d'outils permettant d'aider à leur acquisition et d'un processus d'apprentissage de ces connaissances permettant de les affiner et de les faire évoluer au fur et à mesure de l'utilisation du système.

Les quelques applications du RàPC pour la conception de procédés se focalisent essentiellement sur les connaissances du domaine et utilisent le RàPC dans sa forme la plus basique ou comme un outil d'aide à la décision. Hormis le travail de [AVR 04] sur une mesure de similarité spécifique au domaine, ces travaux ne cherchent pas à adapter les diverses étapes du RàPC aux spécificités du domaine pour créer un véritable outil d'aide à la conception. Pour cela il faut coupler connaissances métiers et expertise en gestion des connaissances.

1.4 Les avantages et les limites du RàPC

D'après [BER 03], le RàPC possède plusieurs avantages, dont la :

- Possibilité de trouver rapidement des solutions à des problèmes complexes.
- Découverte de connaissances, usuellement dissimulées dans les bases de cas, pour la prise de décision.
- Facilité de transfert de connaissances des spécialistes vers un utilisateur non expert.
- Etablissement d'une mémoire collective en partageant différentes expériences.

[P&S 04] et [COR 06] ont également énuméré les avantages suivants :

- *Réduction de la tâche d'acquisition de connaissance.* En éliminant la nécessité d'extraire un modèle ou un ensemble de règles (comme il est nécessaire dans les systèmes basés sur des modèles ou sur des règles). La tâche d'acquisition des connaissances de RàPC consiste principalement à regrouper les expériences existantes, les représenter et les stocker.
- *Éviter la répétition des erreurs commises dans le passé.* Dans les systèmes où les échecs sont stockés aussi bien que les succès, les informations sur la cause des échecs peuvent être utilisées pour éviter des échecs potentiels à l'avenir.
- *Flexibilité dans la modélisation des connaissances.* Les systèmes de RàPC emploient l'expérience antérieure comme connaissance du domaine et peuvent fournir une solution satisfaisante, par l'adaptation appropriée, à ces types de problèmes.
- *Apprentissage avec le temps.* Lors de leur utilisation, les systèmes de RàPC étoffent leur couverture de l'espace des problèmes et des solutions grâce à l'étape « Retenir ». Si les solutions sont validées, ces cas peuvent être ajoutés à la base de cas et être réutilisés pour résoudre des problèmes futurs. Avec cet ajout, un système de RàPC devrait pouvoir raisonner dans une plus grande variété de situations et avec un degré plus élevé de précision et de succès.
- *Raisonnement dans un domaine avec un petit ensemble de connaissances.* Lorsque dans un problème seulement quelques cas sont disponibles, un système de RàPC peut commencer par ces quelques cas et construire sa connaissance au fur et à mesure que les cas sont ajoutés.
- *Production de critères d'évaluation d'une solution.* Au lieu de justifier une solution à un problème, par la démonstration des conditions permettant d'appliquer certaines règles, le RàPC utilise des cas précédents. De ce fait, il est possible de justifier l'utilisation d'une procédure de résolution ou d'une solution par la présentation de cas passés. Ces

justifications sont alors employées comme des critères pour évaluer une solution, lorsqu'il n'y a pas un algorithme disponible ou d'autres critères pour réaliser celles-ci.

Toutefois le RàPC possède également un certain nombre de limites. L'inconvénient principal est la limitation à des problèmes voisins et donc l'absence de garantie d'optimalité des solutions. En effet, la fonction d'évaluation de la similarité est souvent complexe et ne garantit pas que le cas remémoré est bien le cas le plus pertinent. L'efficacité d'un tel système est fortement liée à la couverture de l'espace des problèmes et des solutions, par conséquent la bibliothèque de cas doit contenir un nombre de cas suffisamment important.

L'adaptation du (ou des) cas remémoré n'est pas triviale et demande un certain savoir-faire. Cette phase est d'autant plus complexe que pour un non expert du domaine, les liens ainsi que l'intensité de ces liens, entre les attributs descriptifs du problème et ceux de la solution ne sont pas toujours clairement explicites. Quel que soit le problème *cible* rencontré, un système de RàPC trouvera un cas *source* plus ou moins similaire et donc un tel système est incapable d'établir qu'un problème n'a pas de solution.

Enfin, lors de son utilisation en conception, un système de RàPC manque d'interactivité. En effet, il est intéressant pour le concepteur de pouvoir interagir avec l'outil de façon à visualiser rapidement les conséquences de certains choix sur les autres variables de conception. Afin de réduire les délais de conception, ces choix doivent rester cohérents avec les choix antérieurs. Dans leur forme la plus classique, les systèmes de RàPC ne proposent pas cette interactivité et ne garantissent pas cette cohérence.

1.5 Les applications du RàPC

La complexification des systèmes génèrent un accroissement de la demande en termes de compétences pluridisciplinaires et donc du volume de connaissances à gérer, ce qui amplifie le besoin de la mise en œuvre d'applications issues de l'Intelligence Artificielle. Dans ce cadre, l'approche du RàPC s'est révélée efficace pour résoudre des problèmes dans des domaines très variés allant de la création de recettes de cuisine aux applications industrielles et de services. Les différents systèmes de RàPC ont été classés en six groupes par [MAH 95] en fonction de l'application visée :

- la planification : dans le processus de production d'une séquence d'actions pour atteindre un but spécifique comme la création de recettes de cuisine dans CHEF [HAM 89] et plus récemment dans TAAABLE [BAD 09A] ou WebAdapt [L&P 07] pour la planification

de voyages. Plus proche du domaine du Génie des Procédés, [L&N 96] ont développé un système pour la construction d'un plan de synthèse chimique pour l'obtention d'une molécule cible.

- le diagnostic : le système est alimenté avec un ensemble de symptômes et il essaie de les expliquer, e.g. CASEY [KOT 88] dans le domaine médical, ADAPTER [P&T 95] pour le domaine médical mais aussi celui de la mécanique, KASIMIR [AQU 04] et CABAMAKA [BAD 07] tous deux pour l'aide à la décision en cancérologie du sein, ou dans [HAO 08] qui propose une approche pour le diagnostic technique. D'autres exemples d'applications sur le sujet sont disponibles dans les ouvrages de synthèse [REN 07A], [REN 07B].
- le service client (SAV) : pour assister rapidement et efficacement les clients d'une entreprise, e.g. SMART [A&W 92] créé par Compaq pour aider les usagers de ses produits, CASCADE [SIM 92] développé pour aider les utilisateurs du « Virtual Memory System » face à un échec.
- la classification : le processus d'identification d'un objet ou d'un type de situation, e.g. PROTOS [BAR 89] pour le domaine d'appareils auditifs.
- la configuration : le processus d'identification et d'assemblage des différents composants comme dans CLAVIER [H&H 92] pour le domaine aérospatial ou COMPOSER [P&P 98] pour la séquence de montage et de configuration.
- la conception : l'application du RàPC en conception implique la remémoration de cas de conception et l'adaptation de ces cas afin de répondre aux exigences d'un nouveau problème. La section suivante se propose d'aborder ce domaine d'application.

2 Le RàPC pour le processus de conception

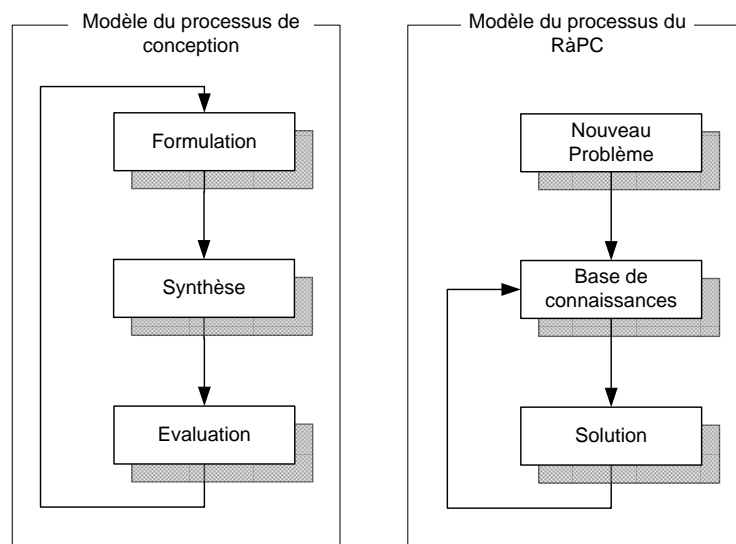
Les systèmes basés sur des règles ou modèles ont été largement développés pour concevoir des systèmes capables d'aider les concepteurs. Bien que de tels systèmes aient connu un succès, des difficultés ont été rencontrées lors de la formalisation des expériences de conception [M&P 97]. A cause de ces écueils, les concepteurs se sont rapidement tournés vers de nouvelles approches telles que le RàPC, pour créer des méthodes et outils visant à faciliter leurs différentes tâches.

Le RàPC est rapidement devenu un outil pour assister les concepteurs lors de la génération de nouvelles solutions, puisque en conception les connaissances expertes et les expériences accumulées dans le domaine sont souvent utilisées pour guider et inspirer de nouvelles solutions [P&P 97]. D'ailleurs, l'idée d'utiliser des techniques de RàPC pour assister

ou automatiser le processus de conception n'est pas récente. Des systèmes comme JULIA [HIN 88], CYCLOPS [NAV 88], KRITIK [GOE 97] ou ARCHIE [G&K 91] furent les précurseurs de ce qui plus tard deviendra l'une des plus grandes branches du RàPC : le « Case-Based Design » (Conception à Partir de Cas) que l'on trouve dans les systèmes comme CASECAD et CADSYM [MAH 95]. Une vaste étude compilant tous ces premiers travaux a été réalisée par [M&P 97].

L'intérêt d'utiliser le RàPC pour assister la conception, est évident lorsqu'on met en regard les deux modèles. Sur la figure 2.3 on peut apprécier leurs ressemblances. D'un côté, on y observe que le modèle du processus de conception se décline en trois actions principales [MAH 95] :

- la formulation : impliquant l'identification des besoins et des spécifications du problème de conception,
- la synthèse : comprenant l'identification des solutions possibles en cohérence avec les spécifications renseignées à l'étape précédente. En effet, elle commence par un ensemble de fonctions et de contraintes de conception qui sont générées dans l'étape de formulation et qui expriment les besoins du concepteur. Par ailleurs le terme « synthèse » est aussi employé pour définir le processus de développement d'une réalisation physique pour une notion abstraite d'un produit [SUL 02].
- l'évaluation : qui interprète et valide les solutions possibles en conformité avec les exigences et performances souhaitées.



Source : adaptée de [MAH 95]

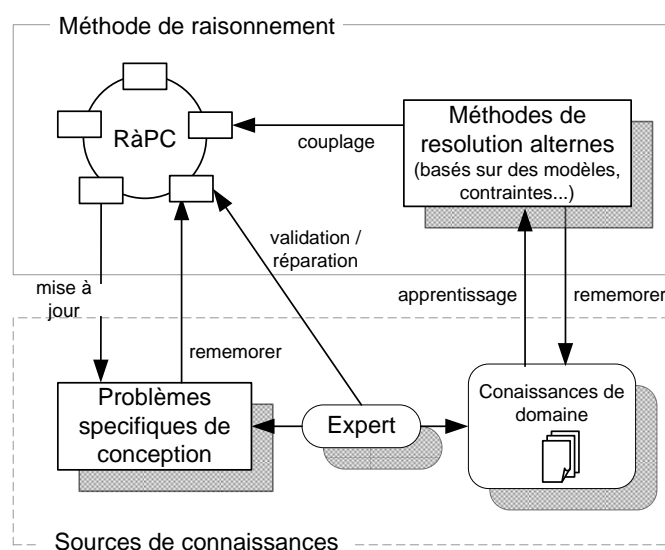
Figure 2.3 Comparaison des modèles de conception et du RàPC

De l'autre côté, le processus du RàPC est résumé par trois étapes principales :

- l'entrée d'un nouveau problème : menant à la formulation et représentation des spécifications d'un problème,
- l'activité liée à la base de connaissances : la recherche dans la mémoire d'un ou plusieurs problèmes semblables qui offrent des solutions potentielles,
- la solution : après évaluation et validation de la solution trouvée (ou adaptée) celle-ci est stockée avec la description du problème dans la base de connaissances pour de futures consultations.

De plus, les deux modèles présentent des étapes à caractère récursif et itératif jusqu'à ce qu'une solution soit finalement validée. Toutefois, un problème est rencontré lorsqu'une solution ne correspond pas aux spécifications du problème *cible* et que des modifications doivent y être apportées pour parvenir à une solution satisfaisante. Le problème s'aggrave encore si les concepteurs sont confrontés à un manque d'expériences et de connaissances sur le domaine d'application.

Face à cette problématique, l'utilisation de nouvelles méthodes pour l'acquisition et l'exploitation des connaissances, à travers le couplage de différentes techniques de résolution de problèmes, peut s'avérer être une option assez intéressante. Les chercheurs du domaine du RàPC se sont penchés vers la mise en œuvre de systèmes hybrides, étendant ainsi la capacité de résolution du RàPC. Un modèle générique d'un système hybride de RàPC en conception est donné sur la figure 2.4.



Source : adaptée de [MAH 95]

Figure 2.4 Structure d'un modèle hybride de RàPC en conception

A noter que pour la formalisation et exploitation de la connaissance, il faut une certaine récurrence dans l'activité. Autant cette récurrence existe en conception demandant peu de changement de rupture, autant elle est moins évidente lorsqu'il s'agit d'aborder une tâche nécessitant une certaine créativité. [EST 06] et [COR 06] ont abordé cette problématique de la créativité dans le RàPC.

2.1 Les avantages du RàPC pour la conception

Le développement de systèmes de RàPC et les avancées de la recherche ont amélioré les performances de cette approche et ont contribué à mettre à la portée des concepteurs, un système capable de :

- suggérer des solutions de conception rapides en diminuant considérablement le temps de résolution et les coûts associés, tout en évitant le temps nécessaire pour concevoir un système à partir de zéro,
- être relativement facile à développer (comparé avec d'autres types de systèmes d'aide à la conception) puisque le processus d'acquisition des connaissances est systématique et peut être bien formalisé,
- servir de base pour créer un pont entre le concepteur et l'ordinateur notamment pour la résolution de problèmes de conception sans limiter le processus créatif [SCH 97],
- interagir avec le concepteur (même si le processus du RàPC tend à être automatique), notamment lorsqu'un échec apparaît, dans des circonstances critiques, et lorsqu'une adaptation est requise,
- offrir un cadre approprié pour l'apprentissage des expériences de conception, sa mise à jour et une exploitation pour des épisodes futurs de conception.

2.2 Le RàPC en conception des procédés

Dans le domaine du génie des procédés, le RàPC n'a eu guère d'application autre que la conception si ce n'est un article récent sur l'estimation des coûts d'investissement pour les équipements des procédés [ERS 12]. Seuranen [SEU 06] dans son étude sur les applications du RàPC en procédés, propose une classification que l'on a étendue en y ajoutant les travaux récents :

- Conception d'opérations unitaires : principalement dans la conception et sélection d'équipements de mélange [KRA 95], pour la conception des échangeurs de chaleur [K&H

97], pour la conception de colonnes à distiller [AVR 04] ou encore pour le choix et le dimensionnement d'internes de colonnes à distiller [NEG 10].

- Conception de procédés : un système pour la réutilisation de flowsheets [S&B 96], dans la conception d'unités de traitement d'eaux résiduaires [AVR 04A], la génération d'alternatives de procédés [LOP 07].
- Conception de procédés de séparation : conception d'un procédé de séparation azéotropique ternaire dans [KIN 99], synthèse d'un procédé de séparation [PAJ 01], exploitation d'une base de modèle MINLP pour la détermination de séquences de distillation [FAR 03], en distillation réactive [AVR 04], dans l'amélioration de la technique séparative du Lit Mobile Vrai [COR 06].
- Conception du système de contrôle : le diagnostic et le contrôle d'un procédé dans la production de papier [XIA 97], dans la conception d'une usine de traitement biologique d'eaux résiduaires [ROD 99] ou pour l'aide à la conception et contrôle d'une usine d'adsorption chimique [ARC 01], à noter également une application en supervision de procédés dans [OLI 09].
- Conception de procédés prenant en compte la sécurité et l'environnement : la conception minimisant l'impact environnemental [KIN 99] ou pour l'évaluation de la sécurité dans la configuration d'un procédé [HEI 98].

3 Contexte et travaux précédents

Cette section offre une présentation sur les fondements et la pratique du raisonnement automatisé pour la conception des procédés. Elle propose une synthèse des diverses contributions développées au cours de ces dernières années au sein de l'équipe de Génie Industriel du département Procédés et Systèmes Industriels (PSI) du Laboratoire de Génie Chimique. On y détaille les travaux portant notamment sur les concepts qui sont au cœur des systèmes de RàPC tels que la mesure de la similarité, la remémoration, l'indexation, l'adaptabilité. Enfin un système de RàPC est construit afin de servir de base pour les contributions des travaux de cette thèse particulièrement sur l'étape d'adaptation.

3.1 La représentation

La plupart des chercheurs en RàPC éprouvent des difficultés sur la structuration, l'identification et la représentation de l'information qui doit être contenue dans un cas. Kolodner [KOL 93] dans ses travaux distinguait déjà deux mesures pragmatiques à prendre

en considérations lors de la construction des cas : la fonctionnalité et la facilité d'obtention de l'information représentée dans le cas. Dans les applications traditionnelles du RàPC, il existe généralement trois types d'approches pour représenter les cas [BER 03] :

- L'approche textuelle : les cas sont représentés sous la forme de textes libres.
- L'approche conversationnelle : les cas sont intégrés par listes de questions-réponses.
- L'approche structurelle : elle s'appuie sur des cas qui sont décrits par des attributs et des valeurs prédéfinies.

Dans [A&K 08], Avramenko et Kraslawski reconnaissent deux manières de représenter les cas avec une approche structurelle :

- Les cas homogènes où chacun des cas appartenant à la base de cas ont la même représentation d'attributs.
- Les cas hétérogènes qui possèdent des attributs différents mais certains peuvent être communs.

De façon alternative, il peut être possible de regrouper les parties communes de plusieurs cas pour en créer un seul de manière artificiel, ou créer une généralisation de plusieurs cas décrivant des problèmes similaires au travers d'un modèle (e.g. ensemble de contraintes). En créant des cas généralisés les aspects les plus importants sont stockés une seule fois dans un cas. Comme on l'expliquera par la suite, une représentation structurelle homogène a été choisie pour le système de RàPC mis en œuvre dans cette thèse.

3.2 Structuration de la base de cas

La remémoration s'appuie souvent sur une structuration de la base de cas dans le but de réduire le temps de recherche et accroître l'efficacité d'identification d'un cas source utile. Une hiérarchie d'index est régulièrement utilisée pour réduire la recherche de cas sources à une portion de la base de cas. Etant donné que le RàPC s'articule autour de l'hypothèse de similarité, il est raisonnable de penser qu'une fois résolu, le problème *cible* sera stocké au voisinage du cas *source* remémoré. En effet, c'est dans l'espace proche du cas *source* que la solution du problème *cible* sera cherchée. De nombreuses méthodes d'indexation se basent sur cette hypothèse pour attribuer un index aux différents cas.

Pour faciliter le processus de recherche, une abstraction du cas peut être construite dans le but de partitionner les attributs en deux classes : les attributs pertinents et ceux qui le

sont moins et qui risquent de bruyter la recherche. A chaque cas est associé un index (considéré en général comme un résumé du cas). Lors de la remémoration, un index est attribué au problème *cible*, qui est ensuite classé dans la hiérarchie d'index des cas sources. Différentes stratégies de construction d'index peuvent être utilisées : la définition des index est faite initialement puis mise à jour automatiquement, la construction de la hiérarchie évolue au fur et à mesure du stockage de nouveaux cas (aspect incrémental du processus). Les index sont repérés par un processus d'apprentissage, utilisation de méthodes inductives pour construire un arbre de décision (détermination des attributs discriminants).

Construire et maintenir une indexation pertinente n'est pas une tâche aisée car elle est fortement dépendante des circonstances de la remémoration qui changent constamment dans les applications réelles. Par conséquent, la structure d'index doit être suffisamment souple pour être facilement maintenue afin d'éviter que de bonnes solutions pour le problème cible ne soient négligées parce qu'elles résident dans une partie inaccessible avec le schéma d'indexation courant. Les conséquences d'une mauvaise indexation peuvent s'avérer désastreuses avec comme conséquences la complexification de l'étape d'adaptation, la réduction de l'efficacité du RàPC, voire l'échec lors de la résolution du problème. De nombreuses méthodes d'indexation ont été développées, il est illusoire d'en dresser une liste exhaustive, [P&S 04] détaillent les principales approches. Toutefois, deux catégories de méthodes émergent :

- Méthodes spécifiques : certains chercheurs proposent des stratégies d'indexation conçues spécifiquement pour leur système de RàPC. Bien que très performantes dans leurs conditions d'application, elles sont difficilement transposables à d'autres systèmes de RàPC.
- Méthodes génériques : la communauté des « machines apprenantes » a largement contribué à développer des méthodes génériques, les arbres de décision en sont le meilleur exemple.

Parmi les nombreuses méthodes, celles sous forme d'arbre de décision sont les plus utilisées dans les systèmes de RàPC. La recherche consiste à parcourir l'arbre et à comparer à chaque nœud les indices, s'il y a appariement le sous arbre est alors sélectionné. L'ensemble des cas est affiné au fur et à mesure de l'évolution dans l'arbre jusqu'à atteindre le nœud final contenant le sous ensemble des cas pertinents (également appelés feuilles de l'arbre).

Dans leurs travaux [PAT 02], [PAT 04] et [G&P 06] ont proposé un type d'indexation générique basé sur une discrétisation de la base de cas (attributs des problèmes discrétisés en

intervalle de taille éventuellement différente). Dans les travaux menés au sein de l'équipe, la méthode d'indexation s'appuie sur un algorithme générique basé également sur une discrétisation de la base de cas. L'algorithme repose en partie sur la méthode d'indexation sphérique proposée par [BRO 08]. La force de cette indexation réside dans le fait que le cas *cible* se localise au centre de la zone de recherche d'une part (on évite ainsi la non sélection de cas sources proches de *cible* mais qui se trouverai dans une autre feuille d'un arbre ou une autre cellule dans les travaux de [PAT 02]) et la possibilité de travailler avec une discrétisation plus fine d'autre part.

L'algorithme d'indexation sphérique consiste à considérer une hypersphère (sphère en 3 dimensions) de recherche autour du cas *Cible* et non plus un hypercube, comme dans les travaux de Patterson, car la taille de l'espace de recherche, reliée au nombre de cellules à considérer, est directement corrélée à l'hypervolume. Par conséquent l'hypersphère est plus avantageuse. En trois dimensions, le rapport des volumes entre une sphère de diamètre D (assimilable à une distance de recherche autour du cas cible) et un cube d'arête de même longueur est de $V_s/V_c = \pi/6 = 52\%$. Ce pourcentage représente la limite basse de l'espace commun entre la sphère et le cube c'est à dire lorsque que le pas de discrétisation tend vers 0 (dans la réalité le gain n'est pas aussi important mais reste substantiel). Pour des bases de plus grandes dimensions, ce rapport décroît drastiquement : 31% en 4 dimensions et autour de 8% en 6 dimensions. Hormis cette réduction de l'espace de recherche, certaines étapes de l'algorithme sont précalculées avant de lancer la recherche ce qui permet un gain en temps de calcul et ainsi de rester efficient malgré une discrétisation plus fine. Sur le critère du temps calcul, 25 tests de recherche ont été réalisés sur un système de RàPC dédié au dimensionnement d'internes de distillation dans le but de comparer l'algorithme d'indexation sphérique et une indexation par R-tree. Les performances sont relativement similaires, avec toutefois un avantage à l'indexation sphérique lorsque la distance de recherche est faible, quand la densité de cas autour du cas cible est forte, mais aussi lorsque la dimension du problème augmente. Cet algorithme est détaillé dans [NEG 10], un résumé est proposé en annexe 1.

3.3 La similarité et ses limites

Tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou sous une autre, un opérateur dont le but est d'établir les ressemblances ou les relations qui existent entre les informations manipulées. Déterminer la similarité entre deux cas est loin d'être une étape triviale, elle

implique l'utilisation de mesures. Dans les systèmes de RàPC, la similarité entre deux cas est établie à l'aide d'une valeur numérique généralement comprise dans l'intervalle [0 :1]. Cette similarité permet d'exprimer que deux problèmes sont similaires à condition que la mesure dépasse un seuil minimum, ou de classer les cas sources en indiquant les plus similaires.

Pour ce qui est des attributs à valeurs numériques, de nombreuses distances ont été proposées mais les plus utilisées étant sans conteste la distance euclidienne et celle dite de « Manhattan » qui ne sont que des cas particuliers de la mesure de Minkowski (pour deux problèmes X et Y) :

$$d(X, Y) = \left(\sum_{i=1}^L w_i |x_i - y_i|^p \right)^{1/p} \quad (2.1)$$

Pour $p=1$ Distance de Manhattan, $p=2$ Distance Euclidienne, $p=\infty$ Distance de Chebychev ($\text{Max} |x_i - y_i|$). Dans la formule précédente, x_i et y_i représentent respectivement les $i^{\text{ème}}$ attributs des problèmes des cas X et Y, et w_i le poids associé à cet attribut. Toutefois, dans de nombreuses études, ce calcul de distance globale n'est pas utilisable car les attributs sont de types différents (non uniquement numérique par exemple), par conséquent une distance pour chaque attribut doit être introduite. De nombreuses mesures de similarité génériques ont été proposées mais une mesure efficace doit reposer sur des connaissances de similarités fortement liées au domaine d'application. Finalement, cette similarité entre deux cas (dite similarité globale) est souvent calculée à partir d'une somme pondérée de similarités entre les attributs (dites similarités locales) décrivant les problèmes.

$$\text{SIM}_{\text{Glo}}(X, Y) = \frac{\sum_i w_i \text{sim}_{\text{loc}}(x_i, y_i)}{\sum_i w_i} \quad (2.2)$$

Pour ces similarités locales, la fonction de similarité décrit mathématiquement comment comparer deux valeurs d'un même attribut. Ces formules de calcul de mesures peuvent intégrer des ontologies métiers, ainsi l'expert peut personnaliser la remémoration selon ses connaissances du domaine. A titre d'exemple, en génie des procédés la nature et la composition du mélange chimique à traiter est souvent un attribut descriptif important. En effet, le mélange conditionne les choix technologiques et influe sur le dimensionnement des appareillages. Calculer la similarité locale entre deux mélanges n'est pas triviale sans ajout de connaissances du domaine. Le système de RàPC, développé au sein de l'équipe PSI, contient

donc des mesures de similarité spécifiques au domaine d'application, l'annexe 2 présente succinctement celle implantée pour les mélanges.

Le système de RàPC élaboré a pour vocation d'apporter une aide en conception préliminaire, par conséquent, la représentation d'un cas en conception doit intégrer la possibilité de décrire et modéliser des connaissances incomplètes ou imprécises. Ces imperfections sur les connaissances sont liées aux difficultés qu'ont les experts à exprimer précisément leurs exigences, mais aussi au manque d'information sur certaines données utiles mais non obligatoires à un instant donné dans le processus de conception. Il est donc important de pouvoir permettre à l'utilisateur de formaliser ces imperfections sans pour autant nuire à l'efficacité du RàPC. Concernant les informations manquantes, le système de RàPC développé possède la fonction « *Ignore* » qui une fois activée, permet de ne pas considérer l'attribut dans la suite du raisonnement.

Pour matérialiser les imprécisions sur les valeurs des attributs, le choix s'est porté vers l'utilisation des sous-ensembles flous car ils offrent non seulement la possibilité de représenter des valeurs numériques imprécises mais aussi de modéliser des expressions linguistiques comme « environ », « peu »... Ces aspects sont détaillés dans l'article [N&L 08B], dont un résumé est fourni en annexe 3.

Le succès et l'efficacité d'un système de RàPC repose sur sa capacité à remémorer des cas pertinents pour la résolution d'un problème, c'est-à-dire des cas facilement réutilisables pour proposer une solution au problème courant. Or retrouver des cas uniquement sur la base d'un calcul de similarité sur la description de problème (appelée similarité de surface) ne garantit pas que le cas source remémoré soit le plus pertinent pour résoudre le problème cible. En effet, le cas le plus similaire au problème rencontré n'est pas forcément celui dont la solution sera la plus facile à adapter. Pour que le système de RàPC puisse prendre en compte cette spécificité, il faut définir un critère sur l'évaluation de l'adaptabilité d'un cas.

3.4 Mesure d'adaptabilité

[S&K 98] ont exploré une autre voie pour faciliter la remémoration de cas pertinents pour la suite du cycle du RàPC. L'idée est d'améliorer l'interprétation de la similarité de surface en lui adjoignant des connaissances supplémentaires sur la signification des attributs. Le principe de leur méthode est de trouver un critère capable de prédire l'adaptabilité d'un cas (avant l'étape d'adaptation), ils ont ainsi introduit la notion de remémoration guidée par

l'adaptation. D'autres approches cherchent à lier similarité et exigences d'adaptation comme par exemple celle de [LEA 97] basée sur l'estimation d'un coût d'adaptation ou celle de [LIE 01] reposant sur une similarité de chemin. Bien que théoriquement très intéressante, dans la pratique ces méthodes demandent des connaissances très spécifiques au domaine d'application (pour le calcul de la fonction de coût par exemple) et par conséquent perdent en généralité.

Sur ce sujet, les travaux de l'équipe, proposent une méthode générique d'évaluation d'un critère d'adaptabilité basée sur l'approche développée par [P&G 07]. Le principe de cette méthode est de lier la facilité d'adaptation d'un cas au potentiel de valeurs solutions que contient son espace d'adaptation. L'idée est de chercher une similarité non plus sur les attributs décrivant les problèmes mais sur ceux de la partie solution. Cet espace d'adaptation est construit à l'aide des attributs de *Sol(source)*, de connaissances supplémentaires et des ensembles flous. L'annexe 4 présente la façon dont l'espace d'adaptation est construit pour un attribut de *Sol(cible)* mais également le calcul de l'adaptabilité de cet attribut. Dans un premier temps, l'évaluation de l'adaptabilité globale d'un cas *Source* se fait par agrégation des adaptabilités locales sur chacun des attributs, ce qui permet de déceler quels sont les attributs d'un cas qu'il sera facile d'adapter et ceux dont l'adaptation risque d'entraîner de fortes modifications.

L'intégration du critère d'adaptabilité dans le cycle de RàPC peut être réalisée de différentes façons, comme par exemple : calculer pour chacun des cas la similarité et l'adaptabilité. Cette solution devient vite inenvisageable dès que la base de cas atteint une certaine taille et/ou que l'évaluation des critères est coûteuse en temps calcul. Une autre possibilité consiste à extraire de la mémoire un ensemble de cas sur la base de la similarité globale puis d'évaluer l'adaptabilité de tous les cas de cet ensemble. L'avantage de cette approche est que la partie coûteuse du traitement se limite à un faible nombre de cas. En contrepartie, ces cas sont susceptibles d'avoir une adaptabilité mauvaise puisque les cas de l'ensemble initial sont choisis à l'aide de la similarité de surface.

Cependant, en conception, il est fréquent de constater que certains éléments de structure sont prépondérants dans la définition des problèmes. D'ailleurs, les concepteurs commencent par identifier les éléments les plus importants avant démarrer la conception. Lorsqu'ils cherchent des cas utiles à la résolution du problème *cible*, ils le font sur la base de ces éléments de structure principaux et non sur la base d'une similarité globale. Ce constat permet de restreindre la sélection des cas sur des attributs prépondérants avant de calculer

les deux critères de remémoration : similarité et adaptabilité. L'apport de l'adaptabilité dans l'étape de remémoration a été testé dans le cadre d'une application dédiée au choix et dimensionnement d'internes de distillation [NEG 10].

3.5 L'exemple d'application du RàPC : le système ReMSiProc

Dans le cadre de l'analyse du RàPC, le système prototype « ReMSiProc » a été conçu pour constituer le domaine applicatif de base pour l'approche axée sur l'étape d'adaptation détaillée au chapitre 4. Dans un premier temps, son développement répond au besoin de tester les différentes approches, parmi celles précédemment étudiées, afin de construire un outil ayant pour but l'assistance dans le domaine de la conception mais également de la simulation en génie des procédés. Cette approche étend les travaux développés à l'échelle de l'opération unitaire, décrits dans les sections précédentes, pour les utiliser à l'échelle du procédé.

Bien que les modèles utilisés pour la simulation de procédés se situent principalement au niveau basique d'ingénierie, selon le cas, la modélisation d'un procédé sous ce niveau, peut devenir très complexe et demander un temps considérable. Pour faire face à ce problème, la plupart des concepteurs tentent de réutiliser des modèles existants, et ajoutent ou suppriment des éléments qui y sont déjà présents. Toutefois, lorsque le nombre de modèles devient important, la recherche d'un modèle similaire peut aussi devenir complexe. Dans ce cadre, le système ReMSiProc offre aux concepteurs un outil capable de rappeler les expériences passées dans le domaine de la conception de modèles, réduisant considérablement le temps de cette activité.

3.5.1 Description de l'outil ReMSiProc

Le système ReMSiProc (**R**emémoration de **M**odèles de **S**imulation de **P**rocédés) est une application du RàPC pour l'aide à la conception de modèles de simulation de procédés chimiques. Il cherche et récupère des modèles de simulation de procédés, stockés dans une base de cas, afin d'en proposer un qui corresponde au mieux aux critères du concepteur.

Il peut éventuellement récupérer d'autres informations associées comme par exemple des analyses environnementales ou de cycle de vie. La base de cas est composée des modèles qui sont des fichiers comportant des éléments tels que le flowsheet, les modèles de simulation, les conditions opératoires, le modèle thermodynamique, précédemment créés

dans des systèmes simulateurs de procédés comme ProSIM [PRO 11] ou COCO [COC 11]. Les principaux avantages de la récupération de modèles sont :

- Proposer des modèles qui puissent être réutilisés afin de réduire leur temps en conception,
- Offrir une ressource qui permette de capitaliser les expériences passées (modèles de procédés) pour assister la tâche de conception.
- La récupération d'autres types d'informations, notamment les résultats de l'analyse du cycle de vie et l'analyse de l'impact environnemental des procédés. En effet, ces informations peuvent être extraites grâce aux contributions faites dans les domaines de l'analyse de cycle de vie menées au sein de l'équipe PSI [MOR 12].

Les modèles peuvent représenter les procédés avec différents niveaux d'abstraction [DOU 88] :

- Niveau conceptuel : comprenant les schémas des fonctions les plus importantes, les équipements et leurs connexions.
- Niveau basique d'ingénierie : comprenant toutes les informations des équipements, connexions, systèmes de contrôle (vannes, capteurs), quelques conditions opératoires (pression, température), composants.
- Niveau détaillé d'ingénierie : incluant toute l'information sur le procédé comme les dimensions, les matériaux, etc.

Le système ReMSiProc se distingue d'autres systèmes de RàPC principalement sur sa flexibilité et sa capacité d'intégration vis-à-vis de l'utilisation des logiciels spécialisés pour la simulation de procédés, en comportant certain indépendance sans le besoin d'établir une façon spécifique de représenter les problèmes. Sur ce dernier point, il se démarque par exemple des travaux de [S&B 96] qui détaillent un système de récupération de flowsheets de procédés. Leur système est principalement axé sur l'étape de représentation des cas. Il présente un inconvénient majeur lorsque l'utilisateur introduit son problème initial. En effet, le système est contraint à l'utilisation d'un vocabulaire spécifique qu'il faut respecter afin d'assurer une mémorisation satisfaisante. De plus, il se focalise plutôt sur l'aspect modélisation, sans prendre en compte l'aspect simulation, et ne comporte pas une étape d'adaptation. Plus récemment, les travaux de [ZHO 10] décrivent un système RàPC capable de réutiliser des modèles de simulation, mais leur système présente l'inconvénient d'être trop spécifique lors de la représentation des modèles, puisqu'ils doivent être organisés d'une façon structurelle. De plus, le système permet seulement traiter des modèles de simulation à événements discrets et le système n'aborde pas l'étape d'adaptation.

3.5.2 Représentation des cas

Dans le système ReMSiProc un cas est un modèle de simulation de procédés défini par un ensemble de paramètres et leurs valeurs. Les paramètres pour l'identification des modèles ont été identifiés grâce à l'avis des experts du domaine et définis dans trois classes principales (figure 2.5) :

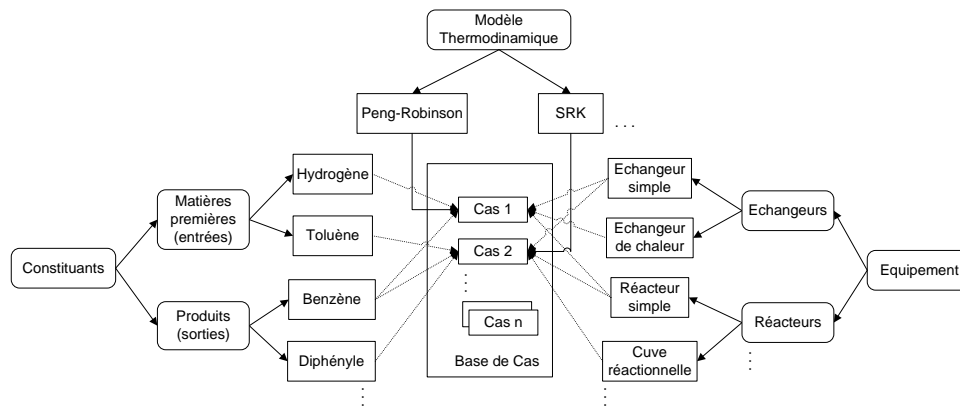


Figure 2.5 La représentation de modèles dans ReMSiProc

- Le type de modèle : indique la principale fonction et le modèle à utiliser pour les calculs thermodynamiques e.g. Peng Robinson, SRK.
- L'équipement utilisé : représente l'ensemble des éléments physiques présents dans le modèle e.g. pompes, réacteurs, colonnes de distillation.
- Les constituants : les produits utilisés dans les réactions, c'est-à-dire les entrées du procédé.

Ces trois classes sont utilisées pour identifier les cas *source* dans la base de cas. La recherche de cas est faite directement dans le dossier contenant les modèles de simulation des procédés. En effet, ReMSiProc est capable de lire dans les fichiers des modèles et de rechercher les informations nécessaires à la recherche de cas similaires. La partie solution est représentée par le modèle de simulation du procédé, son rapport de simulation (e.g. fichiers PDF), et un fichier texte contenant les résultats des analyses de cycle de vie et des impacts environnementaux.

3.5.3 Mesure de similarité

Dans le système ReMSiProc, les cas décrivent, sous la forme de texte, des attributs et valeurs qui peuvent être catégorisés par classes. A l'instar de ce qui a été fait pour les

mélanges de constituants, une mesure de similarité spécifique au modèle thermodynamique a été mise en œuvre. Elle s'appuie également sur une décomposition arborescente.

Dans ReMSiProc les attributs pris en compte pour le calcul de la similarité sont de type texte, le reste de l'information contenue dans les modèles n'étant pas considérée. Ces mots clé représentent les caractéristiques les plus importantes pour l'identification des modèles. Lorsque dans la description du problème il existe des valeurs manquantes, celles-ci ne sont pas prises en compte pour le calcul de la similarité globale. Le pseudo-code de l'algorithme pour ces calculs est présenté sur la figure 2.6.

Algorithme 1 : Calcul de la similarité globale

```

1.  Fonction Lire(cas_cible)
2.  Entier i : 1
3.  Pour tout cas_source ∈ BASE DE CAS faire :
4.  Tant que scanfile(cas_source[i]) != Fin_fichier ;
5.  Si comparaison_string(modèle_cible, modèle_source[i]) == 0 alors
6.  Similarité Locale : modèle++
7.  Pour tout equip ∈ équipement_cible faire :
8.  Si comparaison_string(equip{equip1,..., equipn},
                        equip_cas_source[i]{equip1,... equipm}) == 0 alors
9.  Similarité Locale : équipements++
10. Fin Si
11. Fin Pour
12. Pour tout const ∈ constituant_cible faire :
13. Si comparaison_string(const {const1,..., constn},
                        const_case[i] {const1,..., constm}) == 0 alors
14. Similarité Locale : constituants++
15. Fin Si
16. Fin Pour
17. Calculer similarité global à travers l'équation 2.1
18. Sinon : continuer au cas_source suivant (i++) et retourner au pas (4)
19. Fin Tan que
20. Continuer au cas_source suivant (i++) et retourner au pas (4)
21. Fin pour
22. Retourner Sol(source) : cas_source[i]
23. Fin fonction

```

Figure 2.6 Pseudo-code de l'algorithme de similarité dans ReMSiProc

3.5.4 La remémoration dans ReMSiProc

La représentation des cas et la mesure de similarité introduites précédemment ont été implémentées dans le système ReMSiProc. Le système a été développé dans un environnement graphique rapide pour le développement des applications. L'architecture de ReMSiProc est illustrée sur la figure 2.6.

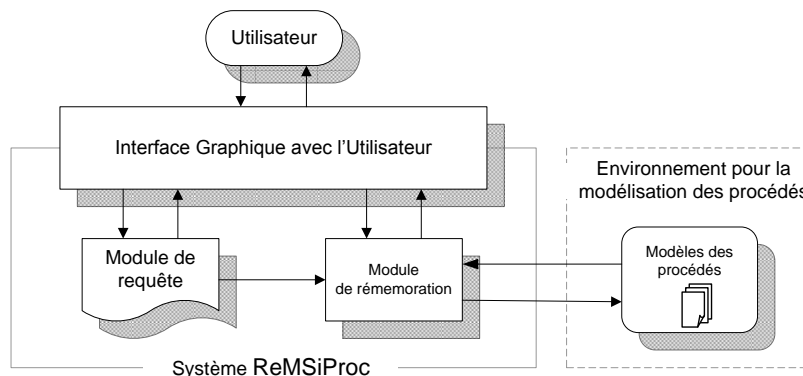


Figure 2.6 L'architecture de ReMSiProc

Sur l'interface graphique le système intègre plusieurs éléments : un module de description de nouveaux problèmes, un module pour la remémoration de cas (algorithme de similarité) et une connexion avec la base de modèles d'un environnement pour la modélisation de procédés. Grâce au module d'entrée de requêtes, l'utilisateur peut introduire facilement un problème à travers différents composants représentant les classes des modèles et listant leurs différentes valeurs. L'interface du module permet de gérer facilement les éléments introduits (ajouter ou effacer).

Après la définition du problème, le module de remémoration lance l'algorithme pour chercher dans la base, des modèles qui soient similaires. Le module de remémoration analyse dans tous les fichiers les classes et attributs saisis lors de la requête. Les résultats de la comparaison entre le problème d'entrée et les modèles remémorés, sont affichés. Finalement en sélectionnant un cas remémoré, le modèle est affiché dans une nouvelle fenêtre :

- pour les modèles de ProSIM plusieurs options sont possibles : une visualisation du rapport PDF, un fichier contenant les informations relatives aux analyses de cycle de vie et des impacts environnementaux [MOR 12],
- les fichiers de Coco pouvant être exécutés depuis l'invite de commandes sont donc visualisés dans leur propre environnement.

3.5.5 L'application de ReMSiProc

Le système ReMSiProc a été testé sur quelques modèles. La base de cas intègre 16 cas. Un exemple sur la production de benzène par Hydrodésalkylation du toluène (HDA) [DOU 88] a été sélectionné pour ce test de par sa relative simplicité. La saisie du problème dans l'interface graphique peut être visualisée sur la figure 2.7.

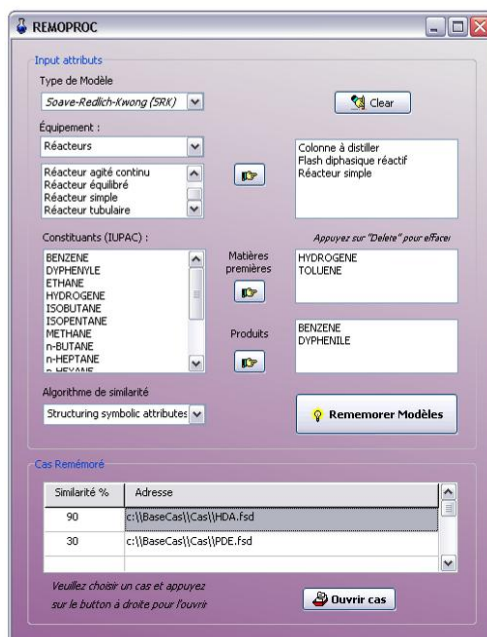
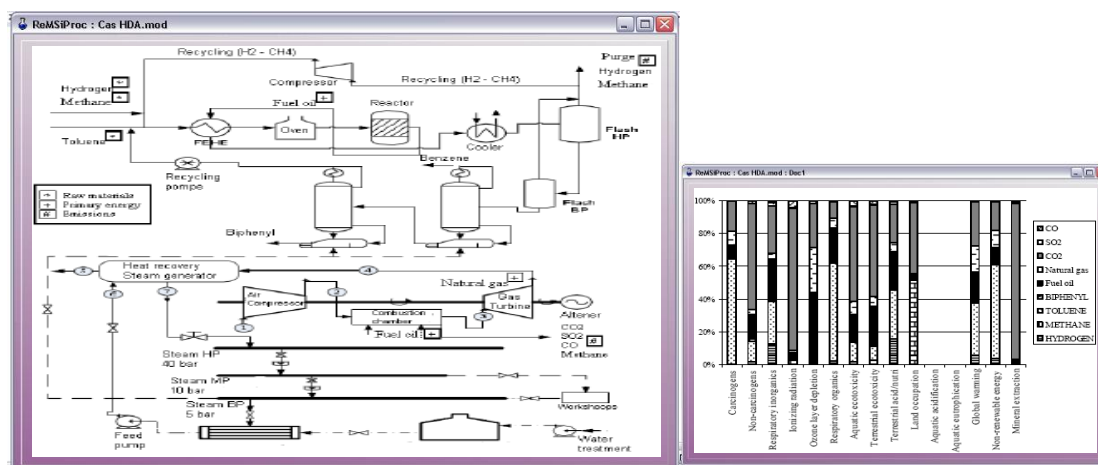


Figure 2.7 Copie d'écran de l'interface graphique de ReMSiProc

Le flowsheet du procédé HDA (comprenant deux réactions : la conversion du toluène en benzène et une réaction parallèle pour obtenir le diphenyle) et les informations concernant les analyses du cycle de vie et impacts environnementaux sont affichées grâce au bouton « ouvrir cas » voir figure 2.8.



Source : [MOR 12]

Figure 2.8 Le modèle de simulation du procédé HDA

Cette vue préliminaire est affichée sur une fenêtre de ReMSiProc, si le système récupère plusieurs cas, chacun d'entre eux peut être aussi visualisé. Une connexion entre ReMSiProc et les outils de modélisation de procédés permet de lancer le modèle à partir de l'interface de ReMSiProc.

4 Conclusion

Les fondements du RàPC ont été présentés dans ce chapitre, ainsi qu'une étude portant sur ses différentes étapes et les travaux menés jusqu'à ce jour. On constate que le RàPC est bien adapté pour assister la conception. En effet, il s'applique bien à des domaines complexes, mal définis, parfois imprécis ainsi qu'à la représentation de connaissances très spécifiques. Les systèmes conçus suivant ce principe peuvent présenter à l'utilisateur des situations spécifiques très contextuelles, proches de la situation à laquelle il doit faire face et l'aider à réutiliser ces situations passées.

L'analyse mise en place dans ce chapitre a permis d'identifier une structure capable d'apporter les moyens pour acquérir, réutiliser et préserver les connaissances pour la résolution d'un problème en conception de procédés. De plus elle a mis en évidence la façon dont les connaissances peuvent être exploitées afin de les capitaliser pour réduire le temps en conception préliminaire. L'outil ReMSiProc a permis de valider l'approche du RàPC classique en mettant en avant l'étape de récupération sans mener aucune adaptation.

Si *source* est suffisamment similaire à *cible* alors $Sol(source)$ est retournée sans modification et utilisée directement comme $Sol(cible)$. En revanche, il est rare que *source* corresponde aux conditions et spécifications de *cible*, il faut alors transformer $Sol(source)$ en une solution appropriée $Sol(cible)$. Les différences entre *cible* et *source* sont d'abord identifiées et $Sol(source)$ est modifiée, tout en prenant compte ces différences. Ce dernier phénomène, intrinsèquement lié au RàPC est défini comme un processus d'adaptation, il sera discuté dans le chapitre 4.

Toutefois, étant donné que l'objectif de ce travail est justement d'améliorer l'adaptation d'une solution trouvée à un problème particulier, ReMSiProc permettra d'établir une architecture pour la mise en place d'une approche visant l'adaptation qui améliorera les performances du système de RàPC. En effet, l'intégration de l'approche par contraintes donnera l'occasion de s'attaquer aux limites du RàPC grâce à une modification du cycle traditionnel de RàPC. L'objectif principal de l'approche proposée est donc de réduire l'effort d'extraction, de formalisation et de modélisation des connaissances.

La Programmation par Contraintes

La programmation par contraintes (PPC), est une approche permettant la résolution d'un large panel de problèmes combinatoires et d'optimisation. Dans ce chapitre on présente un aperçu sur cette technique mathématique et son apport pour la conception. Le but est de l'intégrer au Raisonnement à Partir de Cas pour améliorer l'étape d'adaptation de ce dernier.

Chapitre 3 : La Programmation par Contraintes

1 Principe général de la programmation par contraintes

La Programmation par Contraintes (PPC) est une approche qui a largement fait ses preuves pour la résolution effective de problèmes complexes, particulièrement dans le domaine de l'aide à la décision. La PPC résout des problèmes en employant des techniques de raisonnement et de calcul empruntées à divers domaines comme les mathématiques discrètes, l'analyse numérique, et la recherche opérationnelle.

[WAL 07] définit la PPC comme un paradigme de propagation qui raisonne sur des contraintes. Dans ce cadre, un problème est représenté par un ensemble de variables, chacune ayant un domaine de valeurs associé, et un ensemble de contraintes portant sur ces variables. Ces types de problèmes sont connus comme des problèmes de satisfaction de contraintes (PSC). Résoudre un PSC consiste à trouver une solution dans l'espace des combinaisons possibles. Il existe deux catégories de méthodes de résolution : les méthodes complètes et les méthodes incomplètes. Les premières souvent basées sur une recherche arborescente explorent de façon systématique tout l'espace de recherche et par conséquent attestent d'un résultat : une solution si le problème est cohérent ou la preuve de l'inexistence de solution. Les méthodes incomplètes cherchent une solution de façon heuristique et/ou stochastique, elles n'explorent pas de façon systématique l'espace de recherche (exploration opportuniste de l'ensemble des affectations complètes). Elles permettent de trouver une solution mais sans garantie : par exemple aucune solution n'est trouvée alors que le problème est cohérent et donc pas de preuve d'incohérence du problème. Elles nécessitent une fonction d'évaluation et de comparaison des affectations. Ces méthodes sont généralement mise en œuvre pour résoudre des problèmes de grande taille.

Dans ces méthodes, les contraintes sont utilisées pour tester une cohérence d'affectation (partielle ou complète). A l'inverse, le principe de la PPC est d'utiliser les contraintes durant la résolution pour propager les informations contenues dans une contrainte aux contraintes voisines. Elle applique des techniques de filtrage (ou consistance) qui permettent d'effectuer des déductions et ainsi réduire les domaines des variables. Le mécanisme de propagation de contraintes propage les conséquences d'une affectation ou d'une élimination de valeur à l'ensemble du problème par l'intermédiaire des contraintes. D'après [CHE 07] « *L'objectif de la PPC est de remplacer un PSC initial par un PSC équivalent mais avec un espace de recherche plus restreint* ».

1.1 Applications de la PPC

La PPC a fait ses preuves en tant qu'approche fiable pour la résolution de problèmes très complexes (NP-difficile) d'optimisation combinatoire. Un vaste état de l'art, détaillant les divers domaines d'application de la PPC, est donné dans [FRO 94] qui souligne l'importance de la PPC dans le domaine de l'industrie. Ils citent les applications les plus usuelles suivantes :

- Gestion de projets et d'emploi du temps : planification d'équipes de travail, emploi du temps.
- Gestion et affectation de ressources.
- Optimisation : de production, de moyens, de coûts, de réseaux (informatiques, télécommunications).
- Planification et ordonnancement : planification de la production, d'itinéraires.
- Modélisation : modèles numériques, géométriques, économiques ou financiers.
- Problèmes de configuration et de conception.
- Simulation comportementale de systèmes : le comportement du système à étudier est défini par des relations entre les variables, y compris les entrées et les sorties.
- Vérification de spécifications : le système de résolution de contraintes peut vérifier si le modèle de l'appareil ou du système est conforme à certaines propriétés spécifiées par des contraintes.
- Diagnostic de pannes : face à un comportement anormal, une recherche du composant défectueux peut facilement être mise en œuvre en modélisant le comportement fonctionnel de l'appareil selon des contraintes spécifiques.

Pour étendre encore le champ des applications potentielles de la PPC, des recherches sont conduites pour mettre au point de nouveaux algorithmes et solveurs, afin d'accroître leurs efficacités, leurs fonctionnalités (intégration de divers types de données issues par exemple de code de calcul) [REU 06].

1.2 Les avantages et les limites de la PPC

L'un des plus grands avantages de la PPC est la richesse qu'elle offre en termes de modélisation de problèmes d'optimisation combinatoire. Contrairement à la programmation linéaire, l'expression des contraintes n'est pas limitée à des relations linéaires entre les variables, ce qui offre de grandes possibilités. Un autre avantage de la programmation par

contraintes est de permettre une conduite fine de la résolution d'un problème (notamment d'optimisation), permettant de trouver rapidement une solution [EUR 11]. Parmi les avantages de la PPC on peut citer :

- Une résolution assez rapide de problèmes complexes.
- Une déclaration du problème facile (variable en fonction du nombre de variables et contraintes) et une syntaxe proche de l'expression du problème réel.
- Un travail au niveau des solveurs de PPC réutilisable dans des situations extrêmement diverses.
- La modification, suppression ou l'ajout d'une variable ou contrainte n'implique pas un remaniement complet du modèle initial.
- Le code produit est facilement maintenable et réutilisable.
- Des fonctionnalités de contrôle du processus de résolution au plus près des contraintes.
- Le contrôle du nombre et de la qualité des solutions.

Malgré ces avantages, la PPC présente quelques limites. Par exemple, puisque elle est une méthode de recherche arborescente, qui énumère les différentes solutions du problème, sur certains problèmes de grande taille, ce parcours d'arbre peut devenir très combinatoire et s'avérer très pénalisant en termes de temps de traitement. De plus, la PPC affiche directement les résultats et il est presque impossible de revoir de façon détaillée le déroulement du filtrage pour identifier des éventuelles erreurs [ZAG 11]. Les deux principales limites restant [EUR 11] sont :

- pas de garantie « théorique » sur les temps de recherche,
- pas de garantie de trouver l'optimum, ni d'être en mesure de prouver qu'on l'a trouvé même si tel est le cas. Pour ce faire, il faudrait introduire une fonction objectif et l'évaluer pour chaque solution trouvée.

Cependant, la PPC permet de trouver des solutions en des temps très raisonnables, et en général, les systèmes à base de contraintes ont l'avantage de pouvoir fournir des solutions inédites ou d'établir qu'un problème n'a pas de solution.

2 Aspects mathématiques des PSC

D'après [CHE 07] un PSC « ...peut être traité à l'aide d'un solveur qui calcule un ensemble de solutions, en explorant l'espace de recherche défini par les domaines des

variables, et en s'assurant que les solutions ne violent pas les contraintes formulées ». Dans ce cadre, deux activités principales peuvent résumer la démarche de la PPC :

- La modélisation d'un problème (PSC):
 - L'introduction des variables et la définition de leurs domaines.
 - La modélisation des contraintes selon un formalisme donné.
 - Les modifications possibles sur un modèle PSC déjà existant dans le but d'améliorer sa performance.
- La résolution d'un PSC :
 - Définir ou choisir un algorithme de recherche.
 - Définir ou choisir des heuristiques.
 - L'utilisation d'un solveur de contraintes.

Par la suite, on détaillera ces deux activités principales de la PPC et parallèlement on illustrera le propos à l'aide d'un exemple simplifié sur la configuration d'un réseau d'eau.

Exemple : Configuration d'un réseau Hydraulique

Dans un procédé industriel on voudrait définir un nouveau réseau d'eau d'appoint. Pour ce faire, on dispose de quatre pompes centrifuges de rendement et de type d'impulseur différents (P_1 à P_4), avec au total trois vitesses de rotation différentes (R_1 à R_3). Concernant les conduites, on dispose de trois diamètres (D_1 à D_3) de tubes, ainsi que trois types de matériaux (M_1 à M_3). Des études menées conduisent à un ensemble de contraintes de compatibilités techniques. Tout d'abord, pour des raisons de pertes de charge en sortie de pompe, il n'est pas possible de connecter tous les diamètres de tube sur toutes les pompes : les tubes de diamètre D_1 peuvent se montrer sur les pompes P_2 et P_3 , ceux de diamètre D_2 sur les pompes P_3 et P_4 et enfin ceux de diamètre D_3 sur la pompe P_4 . La vitesse de rotation R_1 est disponible sur les pompes P_1 à P_3 , la vitesse R_3 seulement sur la pompe P_3 . Comme la vitesse de rotation des pompes fixe le débit (puisque les pressions sont supposées constantes) et comme la vitesse de l'eau doit être comprise entre 1 et 2 m/s, ceci génère des contraintes entre diamètre de conduite et vitesse de rotation : la vitesse R_1 n'est compatible qu'avec les diamètres D_1 , la vitesse R_2 avec le diamètre D_2 , la vitesse R_3 avec les diamètres D_2 et D_3 . Au niveau des matériaux, les tubes de diamètres D_2 sont forcément du type M_1 , et ceux de diamètre D_1 et D_3 de type M_2 .

2.1 Les éléments d'un PSC

Le cadre de la PPC est basé sur les notions de variables, de domaines et de contraintes. On s'intéresse donc à trouver une affectation de toutes les variables qui satisfasse toutes les contraintes.

Les **variables**, aussi appelées contraintes inconnues, prennent leurs valeurs dans un ensemble donné, dénommé domaine. En générale elles peuvent être continues ou discrètes. Les techniques de traitement devront tenir compte du type de ces variables. Il est donc possible d'identifier trois types de PSC : discrets (pour des variables symboliques, numériques, entières ou réels), continus (domaines numériques infinis), mixtes (problème combinant les deux types précédents).

Une **contrainte** exprime une relation qui porte sur une ou plusieurs variables qui restreint le degré de liberté lié à l'évolution des variables. Cette relation peut être continue, discrète ou mixte [SAM 95]. C'est une expression reliant des variables de décision. Résoudre ou satisfaire une contrainte consiste à trouver, pour chaque variable du problème, une affectation qui satisfait la condition définie par celle-ci. Ainsi une contrainte provoque une restriction sur les valeurs que peuvent prendre simultanément les variables. Parmi les différents types de contraintes on distingue : les contraintes logiques, arithmétiques, explicites et complexes (e.g. globales⁵ : all-different).

Une contrainte peut être définie en extension (tous les tuples ou l'ensemble des valeurs autorisées), ou en intention (expression synthétique de la restriction sur les domaines des variables). En général on préfère la dernière approche, notamment pour les systèmes de grande taille car elle permet d'exprimer un problème de manière plus concise. Elle est également utilisée pour décrire les PSC numériques, car il n'est pas possible d'énumérer toutes les combinaisons de valeurs possibles pour des intervalles sur les nombres réels.

La **solution** d'un PSC est une instanciation de toutes les variables qui satisfait toutes les contraintes. Elle est une affectation totale et consistante des variables. Si un PSC a une ou plusieurs solutions, alors il est consistant par contre s'il n'y en a aucune on parle d'inconsistante. Selon l'objectif de résolution du PSC on peut chercher :

- une solution, n'importe laquelle, satisfaisant l'ensemble des contraintes,
- l'ensemble des solutions du problème,
- une solution optimale ou au moins une bonne solution par rapport à un critère,

- à prouver la non existence de solution.

2.2 La modélisation d'un PSC

La modélisation consiste à représenter un problème dans un formalisme particulier. On ne se soucie pas de savoir comment le résoudre, on cherche simplement à le spécifier formellement. Cette démarche constitue la principale difficulté pratique de la PPC, car une fois représenté dans le formalisme voulu, un problème peut être résolu automatiquement à l'aide d'un solveur adapté. Un PSC est donc défini par un triplet $\{\mathbf{V}, \mathbf{D}, \mathbf{C}\}$ tel que :

- $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$ est un ensemble fini de variables
- $\mathbf{D} = \{D_1, D_2, \dots, D_n\}$ représente l'ensemble des domaines de valeurs des variables, avec $\forall i=1, \dots, n, x_i \in D_i$. Les domaines peuvent être :
 - Ensembles discrets et finis : $\{1, 2, \dots, n\}, \{2, 3, 5\}, \{\text{rouge, noir, bleu}\}$
 - Intervalles : $[0, k], [1, 2 ; 5, 9]$
 - Valeurs continues : $[3, 4 ; +\infty[,]-\infty ; +\infty[$
- $\mathbf{C} = \{C_1, C_2, \dots, C_n\}$ est un ensemble de contraintes entre les variables, c'est-à-dire un sous-ensemble de $D_1 x_1 \dots x_n D_n$.

En général la modélisation d'un problème comprend plusieurs étapes :

- identification de toutes les variables.
- identification des domaines de valeurs de ces variables.
- identification des contraintes.

Souvent il existe plusieurs modélisations possibles, critères de choix :

- simplicité d'expression
- efficacité : taille de l'espace de recherche de solutions

Exemple : Configuration d'un réseau Hydraulique

Dans l'exemple choisi, la formulation du problème permet de développer un modèle intégré par les éléments fondamentaux d'un PSC. Le problème peut être alors représenté par le triplet $\{\mathbf{V}, \mathbf{D}, \mathbf{C}\}$:

- Variables (V) : Pompes, VitesseR, Diamètre, Matériau
- Domaines (D) : $\{P_1, P_2, P_3, P_4\}, \{R_1, R_2, R_3\}, \{D_1, D_2, D_3\}, \{M_1, M_2, M_3\}$
- Contraintes (C) : C1 (Pompe, Diamètre) ; C2 (VitesseR, Pompe) ; C3 (Diamètre, VitesseR) ; C4 (Matériau, Diamètre).

A noter que les contraintes sont exprimées en extension, et représentées sous la forme d'une macro-structure sur la figure 3.1. La macro structure est un graphe non orienté dont les nœuds sont les variables et les arêtes les contraintes du problème.

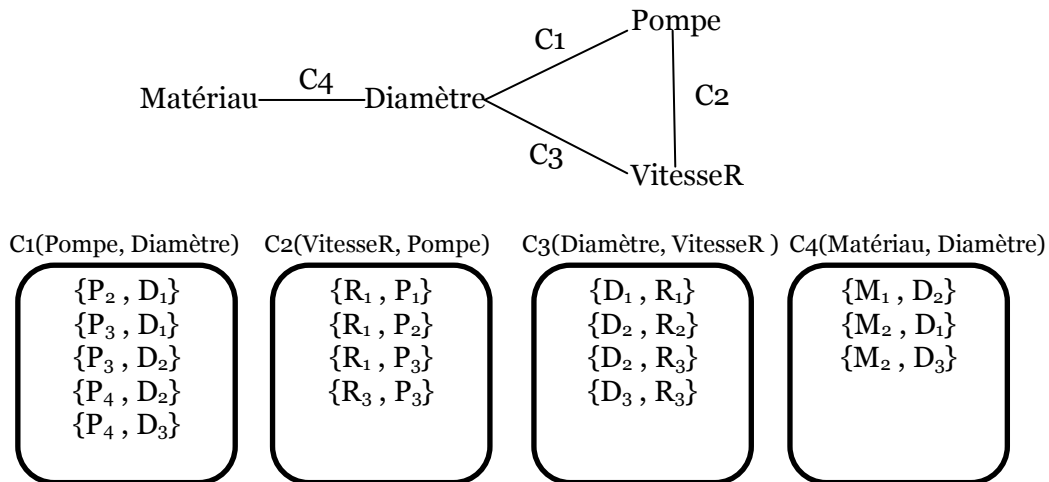


Figure 3.1 Représentation du problème de configuration de réseau

A l'heure actuelle il n'existe pas une méthodologie spécifique pour la modélisation des PSC toutefois on peut énoncer les étapes suivantes [BAI 11] :

- *Partir de la représentation d'une solution* : Il faut d'abord penser à la forme que doit prendre une solution du problème, et ainsi définir les variables.
- *Extraire les contraintes de l'énoncé* : L'énoncé du problème exprime les contraintes ou propriétés qui doivent être respectées par une solution. Il faut que l'expert isole chacune de ces contraintes, puis l'exprime sous la forme d'une relation entre les variables qui ont été préalablement définies.
- *Décomposer certaines contraintes* : Il arrive qu'une des contraintes extraites de l'énoncé du problème ne puisse pas directement être spécifiée dans le formalisme disponible. Il faut alors la décomposer en plusieurs contraintes.
- *Introduire des variables de liaison* : La décomposition de certaines contraintes requiert l'introduction de variables dites de liaison. Ces variables de liaison permettent l'introduction de « valeurs intermédiaires » qui seront utilisées dans de nouvelles contraintes.

2.3 La résolution d'un PSC

La résolution d'un PSC fait généralement appel à deux types d'algorithmes : les algorithmes de recherche et les algorithmes de propagation basés sur des opérateurs de réduction. Ces deux types d'algorithmes sont suffisamment génériques pour s'appliquer à tout type de problèmes.

La propagation de contraintes permet d'inférer des contraintes et des domaines plus restrictifs. Le principe d'une inférence est alors de propager les informations contenues dans une contrainte aux contraintes voisines. Ces inférences sont basées sur l'application de techniques de cohérence de domaines, aussi appelées techniques de consistance, qui permettent de filtrer les valeurs ou les ensembles de valeurs qui participent à une solution. Généralement les techniques employées pour résoudre les PSC dépendent des domaines et sur le type des contraintes utilisées.

Une résolution d'un PSC porte donc sur une affectation de valeurs aux variables de telle sorte que toutes les contraintes soient satisfaites. On distingue plusieurs types d'affectation :

- Affectation complète : affectation de toutes les variables
- Affectation partielle : affectation de certaines variables
- Affectation consistante : affectation qui ne viole aucune contrainte
- Affectation inconsistante : affectation qui viole au moins une contrainte

La résolution d'un PSC est un problème NP-complet dans le cas général. Le mécanisme de résolution consiste (schématiquement) à répéter les deux étapes :

- Réduction des problèmes par filtrage,
- Parcours exhaustive de l'espace de recherche à l'aide d'algorithme de recherche arborescente.

La réduction des problèmes par filtrage, repose sur des propriétés de consistance (ou cohérence) partielle : degré de compatibilité entre valeurs des domaines et contraintes. Différents algorithmes ont été développés pour résoudre les PSC, c'est-à-dire un processus d'instanciation des variables qui permet de déterminer s'il est satisfiable ou pas. Pour la résolution de ces problèmes on distingue trois parties : la consistance locale, les algorithmes de recherche et les heuristiques. Le choix des variables et la manière dont leur domaine est énuméré est souvent crucial pour l'efficacité de l'algorithme de résolution, par conséquent,

des heuristiques peuvent être mises en place. Une brève description des différentes techniques pour chacun des points précédents est donnée par la suite.

2.3.1 Les algorithmes de recherche

La recherche arborescente est l'algorithme de base pour explorer l'espace de recherche de manière systématique. Ils définissent la manière dont est généré et exploré cet arbre. La construction de cet arbre se fait en découpant le domaine d'une variable de manière à obtenir plusieurs sous problèmes à traiter séparément. Dans ces conditions un nœud représente une affectation partielle. Un nœud père et donc un problème équivalent à l'union des problèmes représentés par ses nœuds fils. La racine de l'arbre est une affectation vide, une feuille une affectation complète.

Il existe trois modes de parcours d'arbre de recherche : profondeur d'abord, largeur d'abord et meilleur d'abord (développer le meilleur nœud suivant une fonction d'évaluation). Par rapport aux deux autres modes, celui en profondeur d'abord comporte l'avantage principal d'avoir un espace mémoire requis polynomial. L'algorithme de recherche le plus souvent utilisé est l'algorithme de backtrack.

Les algorithmes de « backtracking » sont la base fondamentale des algorithmes de recherche systématique pour la résolution des PSC. Ces algorithmes ont pour but la recherche dans l'espace des assignations possibles de valeurs aux variables pour trouver une solution, s'il y en a une, ou démontrer que le problème n'a pas de solution.

Dans la recherche arborescente, dès qu'une affectation partielle courante n'est pas consistante alors un mécanisme de retour arrière sur le nœud précédent est opéré afin d'explorer les prochaines branches. La figure 3.2 présente l'exécution de l'algorithme de backtrack pour résoudre le problème de configuration de réseau d'eau.

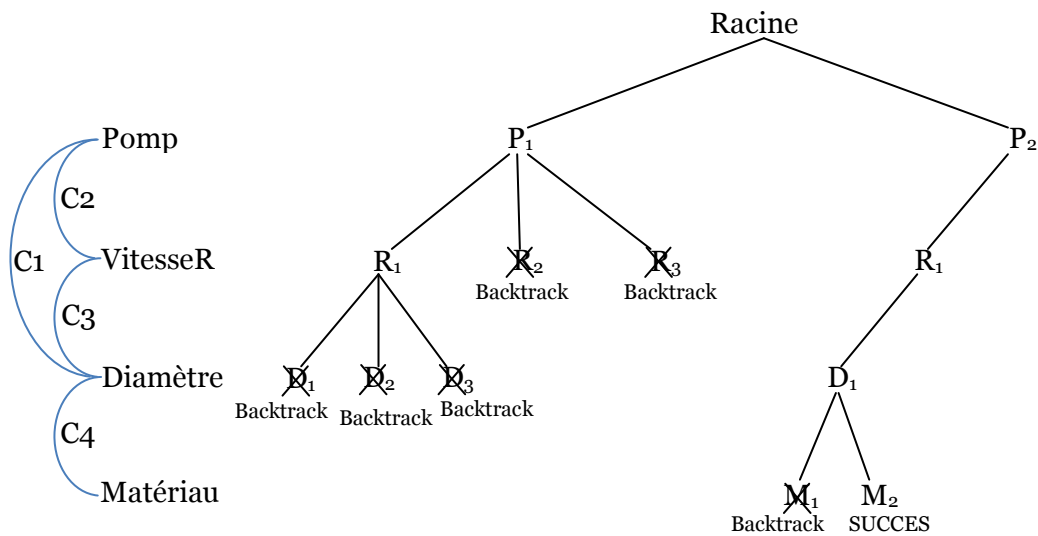


Figure 3.2 Algorithme de Backtracking appliqué à la configuration de réseau

Dans la littérature plusieurs algorithmes de recherche ont été créés, pour les domaines discrets et basés sur des variantes plus ou moins importantes du « Backtrack » comme : « backtraking chronologique », « backjumping » [GAS 79], « conflict-direct backtracking » [PRO 93], « backtracking dynamique » [GIN 93]. A titre d'exemple une amélioration consiste à effectuer les retours arrière de façon plus intelligente. En effet lors d'un échec, on va essayer de remonter à la variable fautive et non pas sur une variable dont on est sûr qu'elle ne joue aucun rôle dans l'incohérence trouvée. La figure 3.3 illustre un tel algorithme pour l'exemple traité. L'algorithme détecte que pour l'affectation partielle (Pompe= P₁; VitesseR= R₁) toutes les extensions suivantes sont incohérentes (contrainte C1 non respectée voire C3 pour certaines valeurs). Donc quelle que soit l'affectation de VitesseR, ses extensions seront incohérentes avec Pompe=P₁. Un Backtrack intelligent revient sur l'affectation de la variable Pompe pour passer à la valeur suivante.

L'une des principales difficultés des algorithmes de recherche est l'apparition d'inconsistances locales qui apparaissent continument [MAC 77]. Les inconsistances locales sont des valeurs individuelles ou des combinaisons de valeurs des variables qui ne peuvent pas participer à une solution puisqu'elles ne satisfont pas au moins une propriété de consistance. Par ailleurs une consistance globale signifie que toutes les valeurs qui ne peuvent pas participer dans une solution sont éliminées.

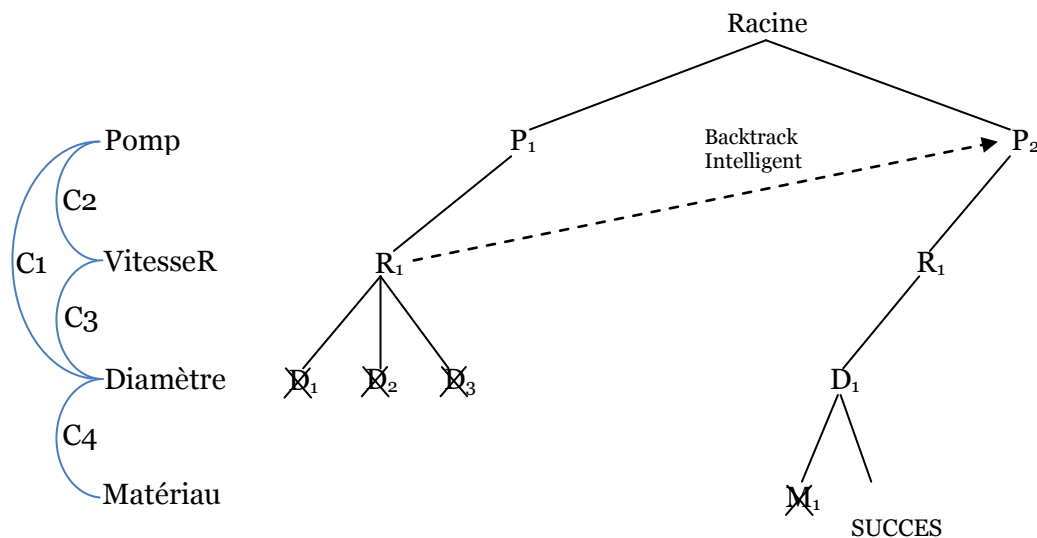


Figure 3.3 Algorithme de Backtrack intelligent

2.3.2 Les heuristiques

Les heuristiques sont des guides pour effectuer des choix, et sont d'une grande importance dans les résultats obtenus en programmation par contraintes [FAG 06]. Un algorithme de recherche requiert un ordre pour l'étude des variables, ainsi qu'un ordre avec lequel on va instancier les valeurs de chacune des variables. Choisir l'ordre correctement des variables et des valeurs peut notamment améliorer l'efficacité de résolution. D'autres heuristiques peuvent tenir compte de la structure du problème et du réseau des contraintes (ordonnement adéquat des contraintes) [S&B 03]. Parmi les différentes heuristiques on trouve notamment des heuristiques sur :

- L'ordonnement des variables : Elles ont été élaborées afin de déterminer la prochaine variable à affecter [BOI 05]. Les heuristiques d'ordonnement des variables essaient de sélectionner, le plus tôt possible, les variables qui restreignent les autres variables. Des heuristiques assez simples sont par exemple choisir en premier : la variable avec le plus petit domaine (adapté pour PSC discret), la variable avec le plus grand domaine (adapté pour PSC continu), la variable la plus contrainte. Sur la figure 3.4 on peut visualiser que le changement de l'ordre de variables réduit le nombre de retour arrière pour l'exemple.
- L'ordonnement des valeurs dans les domaines. Elles changent l'ordre d'exploration des branches de l'arbre de recherche d'un problème. Elles influent fortement sur l'efficacité d'une méthode de recherche [DEC 90]. L'heuristique « min-conflit » [MIN 92] est l'une des méthodes les plus connues. Elle choisit au hasard, une variable instanciée qui est en conflit avec une contrainte quelconque. Ensuite, elle attribue à cette variable la valeur qui réduira le plus le nombre total de conflits. Une nouvelle itération recommence jusqu'à ce

qu'une solution soit trouvée ou un nombre maximal d'itérations soit atteint. Tout comme précédemment, différentes heuristiques peuvent être mises en œuvre en fonction du type des domaines (continu, discret).

- L'ordonnancement des inférences : Elle est effectuée par l'insertion des révisions à effectuer dans une structure particulière de données. Elle est aisément implémentée en traitant d'abord les révisions de contraintes issues des dernières suppressions. Dans le cas de problèmes inconsistants, il a été observé que ces révisions aboutissent plus rapidement à une solution [FRE 95]. D'une façon plus générale Wallace et Freuder [W&F 92] ont proposé différentes heuristiques d'ordonnancement de ces inférences. Les plus performantes choisissent dans la structure stockant les arcs à réviser, l'arc dont la contrainte possède la plus grande complexité, ou bien encore elles sélectionnent l'arc dont les variables ont les plus petits domaines [BEN 10].

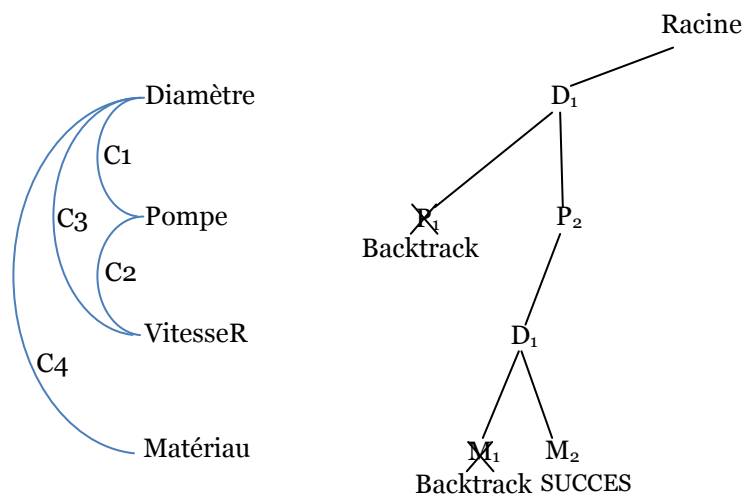


Figure 3.4 Backtrack avec heuristique sur ordre des variables

2.3.3 Filtrage sur des domaines discrets

Dans la littérature il a été proposé plusieurs techniques de consistance locale afin d'améliorer l'efficacité des algorithmes de recherche. De telles techniques éliminent des valeurs inconsistantes des domaines des variables ou induisent des restrictions implicites qui aident à réduire l'espace de recherche. Ces techniques de consistance locale sont utilisées avant de commencer ou pendant la recherche afin de réduire l'arbre de recherche.

Dans les techniques de résolution précédentes, les contraintes sont utilisées que pour tester la cohérence d'une affectation (utilisation passive). Les techniques de filtrage utilisent les contraintes de manière active pour effectuer des déductions sur le problème. L'objectif principal de ces techniques est de détecter des affectations partielles incohérentes, et ainsi de

réduire les domaines des variables en s'appuyant sur les contraintes. Une des techniques les plus utilisées est la cohérence d'arc. Cette dernière vérifie que toute valeur du domaine d'une variable est compatible avec chaque contrainte prise séparément.

Exemple : Configuration d'un réseau Hydraulique

Un exemple d'application du filtrage par arc cohérence au problème exposé dans l'exemple du réseau d'eau simplifié le problème initial à :

$$D(\text{Pompe}) = \{P_2; P_3\}; D(\text{diamètre}) = \{D_1; D_2\}$$

$$D(\text{Vitesse R}) = \{R_1; R_3\}; D(\text{Matériau}) = \{M_1; M_2\}$$

Il existe de nombreux algorithmes différents implémentant la cohérence d'arc [LEC 03A], [LEC 03B] et [BES 06] en donnent un descriptif. En effet ils offrent un bon compromis entre force du filtrage et temps calcul. Le filtrage idéal supprime toutes les valeurs incohérentes de tous les domaines (filtrage global) mais le temps calcul devient rédhibitoire. Le degré de filtrage correspond donc au nombre de variables k participant à la vérification de la cohérence locale (par opposition à filtrage global), on parle de k -consistance. Si $k=2$, la 2-cohérence équivaut à la cohérence d'arc. Pour $k>2$, les algorithmes s'appliquent uniquement pour des contraintes binaires contrairement à la cohérence d'arc qui a été étendue à des contraintes n -aires. En général, les k -consistance (pour $k>3$) ont une complexité trop élevée pour une utilisation dans la plupart des problèmes.

Dans la pratique, les techniques de filtrage sont couplées avec des techniques de recherche arborescente. Une façon d'effectuer le couplage est de réaliser le filtrage pendant la recherche arborescente. A chaque nœud constituant un sous problème on effectue un filtrage. L'algorithme de « Forward Checking » combine un Back Tracking avec une anticipation par arc-cohérence. Cette dernière permet d'éliminer les valeurs des variables non encore instanciées qui sont incompatibles avec l'affectation partielle courante. Si un domaine d'une variable non instanciée devient vide alors l'affectation n'est globalement pas cohérente et un retour-arrière est effectué. La figure 3.5 démontre l'application de cet algorithme à l'exemple. L'affectation partielle $\text{Pompe}=P_1$ permet de réduire le domaine des variables VitesseR et Diamètre. Le domaine de cette dernière variable étant vide prouve que cette affectation partielle ($\text{Pompe}=P_1$) n'est pas globalement cohérente. Avec cet algorithme on n'effectue qu'un seule Backtrack sur l'exemple. A noter qu'il existe d'autres algorithmes que le Forward-checking, on peut notamment citer l'algorithme Look-ahead.

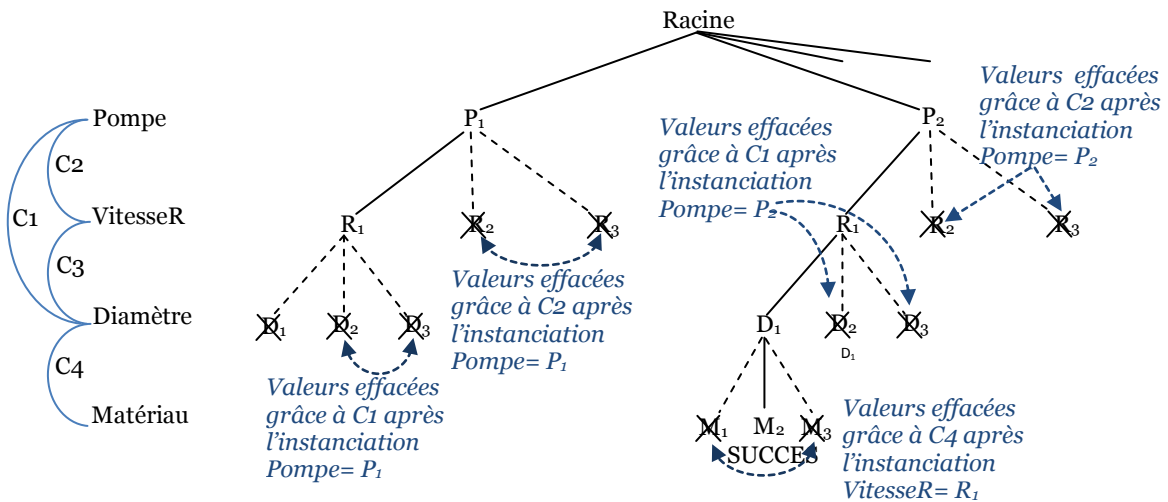


Figure 3.5 Algorithme de Forward Checking appliqué à la résolution de l'exemple

2.3.4 Le filtrage sur les domaines continus

D'après [BEN 06] le traitement des domaines continus peut se faire suivant deux approches : par une discrétisation des domaines continus ou sur la base du calcul ensembliste. Pour cette dernière, diverses techniques mathématiques (équivalentes des méthodes de filtrage sur domaines discrets) ont été développées permettant de réduire les domaines continus afin de calculer des solutions consistantes. On appelle ces types de problèmes des PSC numériques. [DEL 00] a réalisé une classification des différentes méthodes de filtrage pour ces problèmes et [VAR 05] les compare en termes de puissance et degré de filtrage. Chacune de ces méthodes est plus adaptée à certaines formes de contraintes numériques. Pour des fonctions continues monotones, la 2B-cohérence procure un filtrage suffisant. Pour des fonctions non-monotones (mais dérivables), la Box-cohérence est plus adaptée [OUD 06].

A noter que la recherche de toutes les solutions sur des PSC numériques pose des difficultés (mathématiques et de temps de calcul) notamment lors de l'identification de tous les éléments d'un ensemble infini. La recherche de solutions est complète et exacte pour les PSC discrets (problème NP-complet) alors qu'elle est approchée pour les PSC numériques [OUD 06]. En effet, les techniques permettant de les résoudre reposent sur l'arithmétique des intervalles.

2.3.4.1 L'arithmétique des intervalles

Les bases de l'arithmétique des intervalles ont été établies dans les travaux de [MOO 66]. Celles-ci correspondent aux quatre opérations de base étendues aux intervalles. Soient $[a; b]$ et $[c; d]$ deux intervalles :

$$[a; b] + [c; d] = [a + c; b + d] ;$$

$$[a; b] - [c; d] = [a - d; b - c] ;$$

$$[a; b] * [c; d] = [\min(ac, ad, bc, bd) ; \max(ac, ad, bc, bd)] ;$$

$$[a; b] / [c; d] = [\min(a/c, a/d, b/c, b/d) ; \max(a/c, a/d, b/c, b/d)] \text{ si } 0 \notin [c; d],$$

Plus tard [NEU 90] étend cette arithmétique aux fonctions élémentaires (e.g. cos, log) et [SHA 98] quant à lui, démontre qu'en l'étendant aux multi-intervalles, il n'y a pas d'explosion combinatoire. Dans cette dernière, les opérations de base deviennent les unions des opérations de base sur les produits cartésiens des intervalles. Dans l'exemple ci-dessous proposé par [OUD 06], une contrainte est étendue aux intervalles.

Exemple : Extension d'une contrainte aux intervalles

Soit a, b, c, d et z des variables continues définies sur :

$$D_{a,b,c,d} = [1 ; 2] \text{ et } D_z = [0 ; 10]$$

Soit la contrainte : $z = 2 \times a + b/c - d$

$$\text{Alors } z = (2 \times a) + (b / c) - d$$

$$= (2 \times [1 ; 2]) + ([1 ; 2] / [1 ; 2]) - [1 ; 2]$$

$$= [2 ; 4] + [0,5 ; 2] - [1 ; 2]$$

$$= [2,5 ; 6] - [1 ; 2]$$

$$= [0,5 ; 5]$$

D'un côté, l'approche de satisfaction de contraintes par intervalles offre la possibilité de [HYV 94] : représenter des problèmes inexacts en termes d'intervalles, calculer des solutions à un niveau plus abstrait, raisonner dans des situations sous-contraindantes et avec des données inconsistantes et représenter des solutions inexactes et généralisées. De l'autre, le filtrage par l'arithmétique des intervalles [OUD 06] : est facile à mettre en œuvre (il suffit de définir les opérateurs mathématiques) et permet de ne pas trop alourdir les calculs, garantit de ne pas perdre de valeurs pouvant mener à des solutions et permet de filtrer plus ou moins efficacement tout type de fonction (pas d'hypothèses d'application contraignantes). Les points suivants décrivent deux des techniques les plus connues sur les intervalles.

2.3.4.2 2B-cohérence et Box-cohérence

La 2B-cohérence (Hull cohérence) est définie comme un filtrage aux limites des intervalles [LHO 93]. Une contrainte est donc 2B-cohérente si pour chacune des variables qu'elle englobe prenant la valeur des bornes de son intervalle de définition, il existe au moins une valeur dans le domaine des autres variables permettant de satisfaire la contrainte. La 2B-cohérence permet de ne pas perdre de valeurs pouvant mener à des solutions. Cependant, les hypothèses d'application de cette méthode de filtrage numérique sont trop contraignantes [OUD 06]. Pour pouvoir appliquer la 2B-cohérence, les contraintes numériques doivent être des fonctions : sans multi-occurrence de variables (celles-ci n'apparaissant qu'une seule fois dans chaque contrainte), monotones et projetables⁶ sur chacune de leurs variables. D'après [OUD 06], en conception, ces trois conditions sont rarement remplies à cause de la complexité des problèmes.

La Box-cohérence se propose d'aller au-delà des conditions d'application de la 2B-cohérence [BEN 94]. Le but de la Box-cohérence est de déterminer le domaine le plus petit possible par réduction successive des bornes supérieures et inférieures du domaine de définition [SAM 95]. Une contrainte est Box-cohérente si pour toute variable X_i appartenant à cette contrainte, les bornes de l'intervalle de la variable X_i vérifient la contrainte obtenue en remplaçant chaque occurrence de X_j (X_j , X_i) par l'intervalle de valeurs de X_j .

D'après [CHE 07] face à des contraintes complexes (arité supérieure à trois), la Box-cohérence permet d'obtenir une meilleure approximation que la 2B-cohérence sur l'ensemble de solutions. En conception préliminaire cela semble avantageux car souvent les modèles comportent des contraintes complexes. [CHE 07] propose une bonne alternative en combinant ces deux consistances afin d'obtenir de meilleurs résultats vis-à-vis du temps calcul.

3 La PPC en conception

La PPC propose une façon naturelle de représenter les problèmes puisque l'utilisateur n'a plus qu'à énoncer les contraintes du système à modéliser. La difficulté est de pouvoir exprimer les restrictions relatives à la conception d'un système. L'avantage de la formulation de problèmes comme des PSC est que c'est un mode de raisonnement qui fournit un même cadre pour la modélisation et la résolution d'un problème. Le PSC fournit une manière très simple et avantageuse de représenter des problèmes de par son approche déclarative lors de

⁶ La projection consiste à éliminer du domaine D_x les valeurs « localement » incohérentes vis-à-vis de la contrainte c [CHA 07].

la modélisation [SQA 99]. Cet aspect déclaratif du problème sous forme de relations simples rend la modification de la base de connaissances facile, car il est toujours possible d'ajouter ou d'enlever une contrainte [HOK 06]. Mais l'un des avantages les plus importants de l'approche de PSC est qu'il suffit d'une formulation appropriée pour la résolution [SUL 02]. La PPC est également indépendante du domaine d'étude grâce à son niveau d'abstraction. Lorsqu'un système est représenté comme un PSC, il peut être résolu indépendamment du contexte ou du domaine initial de l'application.

3.1 L'intérêt du formalisme PSC en conception

Dans les travaux de [OUD 06] on identifie deux types d'assistance pour l'aide à la conception : autonome ou interactive. Selon lui, dans la conception autonome il s'agit de « *...trouver une ou toutes les solutions acceptables à partir d'un modèle générique (éventuellement augmenté de certains choix de l'utilisateur ou des évaluations de variables, par exemple)* », tandis que dans la conception interactive on retire du domaine des variables les valeurs qui ne sont plus cohérentes avec les contraintes (ou les choix de l'utilisateur). A noter qu'en distinguant ces deux types de conception (autonome et interactive) on différencie respectivement les deux techniques de traitement des PSC : la résolution, lorsqu'il s'agit de trouver solutions consistantes ou le filtrage si l'on veut réduire les domaines des variables en tentant d'y supprimer les valeurs qui n'interviennent dans aucune solution.

D'après ces principes, une résolution basée sur les techniques de PPC permet de calculer l'ensemble des solutions d'un problème. Par opposition à d'autres méthodes qui se limitent à calculer des solutions optimales ou un certain nombre d'entre elles. Dans le domaine de la conception, l'intérêt se pose dans la résolution sur des domaines continus. Les PSC offrent la certitude sur la non existence de solutions et par conséquent les concepteurs bénéficient d'une meilleure perception de l'ensemble des solutions du problème.

De plus, la dissociation des étapes de modélisation et de résolution permet aux concepteurs de se focaliser sur la description de son problème suivant le formalisme des PSC. Les modifications des PSC sont plus faciles à gérer. Selon [CHE 07] « *...la conception est un processus dynamique, ou les modifications apportées au modèle peuvent être fréquentes et plus particulièrement lors de la conception préliminaire, ou peu de décisions ont encore été prises. Certaines décisions peuvent avoir lieu pendant cette phase et nécessiter d'importants changements au sein du modèle (choix ou changement de composants, de technologie, de topologie, etc.)* ». D'ailleurs, cette séparation de la résolution facilite la réutilisation de la

connaissance sous le formalisme PSC. Les modèles PSC peuvent être intégrés dans un formalisme plus général et de plus haute abstraction, permettant aux concepteurs de définir la connaissance plus exhaustivement et faciliter sa manipulation [CHE 07].

Pour éviter des solutions non réalisables, il est intéressant de recenser les alternatives possibles afin d'éviter les échecs d'une part et de limiter les itérations dans le processus de conception d'autre part (atteinte d'une solution en un temps acceptable). Garder la cohérence entre une décision prise et les décisions antérieures participe également à la réduction du temps de conception. De plus cette approche offre une certaine interactivité puisque les concepteurs pourront voir les conséquences ainsi que la cohérence d'un choix sur les autres variables de conception. La difficulté majeure pour l'application des PSC en conception concerne l'effort conséquent d'analyse et de formalisation des connaissances lors de l'étape de modélisation. A titre d'exemple, cette étape est plus longue et plus complexe que son équivalente dans le RàPC.

3.2 La PPC et le processus de conception

Les techniques de la programmation par contraintes ont été appliquées dans plusieurs domaines en conception durant ces dernières années. L'un des premiers travaux a été mené par Serrano [SER 87] dans le domaine de la gestion de contraintes pour la conception en ingénierie grâce à son outil « Concept Modeller ».

Plus tard, en se basant sur ce système, Buckley [BUC 92] et Reddy [RED 96] concevaient un outil appelé « Design Sheet » qui permettait aux concepteurs de créer des modèles de conception en utilisant un ensemble de contraintes algébriques. D'autres chercheurs comme Gelle et Smith [G&S 95] ou Haroud [HAR 95] utilisaient l'approche de PSC dynamiques comme base pour résoudre des conflits pendant la phase de conception préliminaire. D'autres ont également proposé des systèmes collaboratifs pour la gestion de conflits pendant la conception [BAH 94].

Plus récemment certains travaux proposent l'utilisation de contraintes pour assister les concepteurs comme dans [B&D 06A] utilisant l'outil Datalog ou dans [VAR 07] qui décrit un outil pour assister la conception basée sur contraintes en utilisant une fonction pour évaluer les solutions. [WIT 09] proposent un outil pour modéliser et résoudre des problèmes de configuration et [BIN 10] utilisent la formulation de PSC pour la génération d'essais de

programmes. La figure 3.6 montre une comparaison entre le modèle de conception et l'approche par contraintes. Ce dernier est constitué autour de trois actions principales :

- une formulation comprenant variables, domaines et contraintes,
- une résolution conduite grâce à un algorithme de recherche ou inférence,
- la solution soumise à une évaluation.

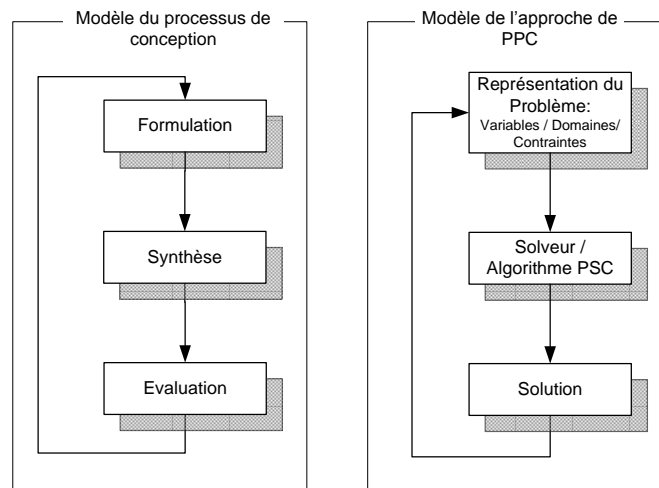


Figure 3.6 Comparaison du modèle de conception et de l'approche par contraintes

Les différents avantages de l'utilisation de l'approche par contraintes ont été donnés dans les points précédents. Toutefois l'une des limites la plus importante est que dans beaucoup d'applications la distance entre le modèle et le programme par contraintes peut être encore longue, principalement en raison des particularités des langages de PPC qui peuvent, eux aussi, demander un certain niveau d'expertise [HEN 99].

Une autre limitation concerne l'échec qui intervient à l'issue d'une propagation de contrainte lorsque le domaine d'une variable devient vide. Face à un tel échec, le concepteur revient sur ses choix antérieurs pour en déduire la cause. Souvent, le concepteur se focalise sur son dernier choix, mais, dans des contextes particuliers, tous les choix antérieurs peuvent avoir une influence [HAD 02].

De même si le concepteur est démuné de connaissances ou si elles sont mal définies ou imprécises comme c'est souvent le cas en conception préliminaire, des sources alternatives de connaissances sont requises pour remédier à ce problème. Une coopération entre la connaissance générale représentée par les modèles de PSC et la connaissance contextuelle (e.g. RàPC) peut être établie afin de compléter les deux approches et proposer une solution satisfaisante aux concepteurs [VAR 11].

3.2.1 Le modèle PSC en conception

Les applications réelles en conception impliquent l'utilisation de variables discrètes et continues [G&F 03]. Pour cette raison l'utilisation de variables sur les domaines continus est devenue essentielle dans la PSC ; surtout en conception où souvent, d'après [CHE 07] « ...*les problèmes définis sont alors continus ou mixtes : les variables discrètes représentent certains choix de valeurs normalisées ou de composants et les variables continues expriment certaines dimensions ou certaines propriétés des phénomènes physiques* ».

En conception, le modèle de PSC met en évidence plusieurs types de variables et contraintes, comme résultat du besoin lié à la modélisation. Selon [HOK 06] le but est de « ...*pouvoir différencier des configurations de conception les unes par rapport aux autres, en termes de performances et de coût, afin d'aider au choix.* » Dans les travaux de [VER 04], deux types de variables sont discernés afin de traduire les caractéristiques structurantes identifiées lors de la phase d'analyse ([CHE 07] les identifie comme variables de décision et variables alias) :

- **les variables de conception** : qui d'après lui : « *définissent de manière stricte le composant auquel elles sont associées (longueur, couleur, référence dans un catalogue). Une fois évaluées, elles permettent donc de distinguer deux configurations de conception entre elles ;* »
- **les variables auxiliaires** : qui correspondent aux autres variables, c'est-à-dire qui ne sont pas des variables de conception. Elles sont nécessaires à la modélisation pour relier mathématiquement les variables de conception aux critères (débit, effort extérieur, pression, nombre adimensionnels). Ce lien est réalisé au travers de l'expression des contraintes.

Du côté des contraintes, plusieurs travaux, notamment dans [G&S 95], [GEL 98], [GEL 00] et [G&F 03], abordent le problème lié à l'utilisation des contraintes mêlant des variables discrètes et continues. Dans ces travaux ils définissent les types de contraintes vis-à-vis de la conception :

- **les contraintes discrètes** : sont basées sur la représentation des combinaisons de valeurs discrètes, soit en énumérant explicitement des tuples, soit par le biais de formules logiques.
- **les contraintes continues** : utilisent des représentations implicites de points permis dans un espace continu sous la forme d'une représentation analytique.

- **les contraintes mixtes** : sont des relations définies mêlant des variables discrètes et variables continues (numériques).
- **les contraintes conditionnelles** : définissent des conditions sous lesquelles une variable doit exister ou non. Par exemple, elles expriment lorsqu'un ensemble de composants nécessitent un autre composant dans un contexte spécifique.

D'autres auteurs s'intéressent à l'utilisation des contraintes globales afin de les adapter aux problèmes de conception, comme dans [CHE 07] où la contrainte globale « *piecewise* » est revisitée pour l'appliquer à des fonctions non linéaires. Vu que la plupart des phénomènes physiques complexes sont perçus au travers de variables pertinentes (e.g. nombre de Reynolds en mécanique des fluides), c'est grâce à l'utilisation de ce type de contrainte « *piecewise* » que les modélisations plus complexes deviennent donc facilement gérables dans les PSC.

Exemple : La contrainte globale « *piecewise* »

Lorsque on analyse des phénomènes de dégradation d'énergie dans un fluide, différents types d'écoulement (i.e. laminaire, transitoire et turbulent) sont identifiés par le nombre adimensionnel de Reynolds (Re). Le facteur de Fanning (f) lié aux frottements peut être exprimé de plusieurs manières selon la valeur du nombre de Reynolds.

$$Re \in [100 ; 2100] \rightarrow f = 16/Re$$

$$Re \in [2100 ; 5 \times 10^4] \rightarrow f = 0.10512 \cdot Re^{-0.244}$$

$$Re \in [5 \times 10^4 ; 10^6] \rightarrow f = 0.04234 \cdot Re^{-0.164}$$

La contrainte globale « *piecewise* » peut être donc définie de la façon suivante :

$$piecewise (Re, [100 ; 2100] : f = 16/Re ; [2100 ; 5 \times 10^4] : f = 0.10512 \cdot Re^{-0.244} ; [5 \times 10^4 ; 10^6] : f = 0.04234 \cdot Re^{-0.164})$$

3.3 La programmation par contraintes en procédés

Dans le domaine de la conception de procédés, la PPC n'a connu que très peu d'application. Parmi les premiers travaux, ceux de Gelle [GEL 98] proposent une modélisation de connaissances axée autour de la problématique d'aide à la conception de mélangeurs industriels où le besoin de lier des variables discrètes (géométrie de la cuve du mélangeur) à des variables continues (son volume) est abordée. Cet exemple reste sommaire

puisque très peu de types d'agitateur, de géométrie de cuve sont intégrés dans le modèle. L'exemple traité sert plus à la validation de l'approche mathématique qu'à une réelle application en génie des procédés.

[ABB 99] utilisent l'approche de la programmation logique par contraintes (PLC) pour résoudre le problème de la conception « rétrofit⁷ » de réseaux d'échangeurs de chaleur. L'approche combine l'intuition des experts, représentée par une série d'heuristiques par la programmation linéaire, et une technique d'optimisation.

Dans [S&G 99] l'approche de PSC dynamique⁸ permet une représentation de connaissances plus expressive et généralisée surtout pour les problèmes de configuration. Plus tard dans [G&F 03], une approche est proposée pour résoudre des PSC mixtes avec des contraintes conditionnelles toujours dans le domaine de configuration de mélangeurs industriels. Toutefois, ces méthodes sont basées sur une discrétisation des domaines continus et ne peuvent pas garantir la cohérence globale du PSC mixte [OUD 06].

La PPC est souvent utilisée en tant que technique mathématique dans le cadre de l'optimisation de systèmes complexes. Elle sert à restreindre le domaine des variables d'optimisation. Par exemple, [J&G 01], [H&G 02] et [M&G 04] utilisent des méthodes de résolution de problèmes MILP couplées avec la PPC. Ces approches aussi appelées « hybrides » permettent de résoudre des problèmes d'ordonnement de procédés « batch » grâce à une assignation des décisions gérées par un sous-problème MILP et des séquences de décisions traitées par un sous-problème PSC [MEN 05].

[HOK 06] de son côté, s'appuie sur une démarche de formalisation de modèles portée sur le formalisme des PSC pour développer un outil d'aide à la décision en conception préliminaire d'évaporateurs flash bi-étagés.

Finalement dans [OUD 06], une contribution à l'élaboration d'un formalisme gérant la pertinence pour les problèmes d'aide à la conception à base de contraintes est menée grâce à l'approche de RCSP (Relevancy Constraint Satisfaction Problem). L'application est faite sur un problème de conception d'un réacteur chimique.

⁷ Il consiste à échanger des parties ou composants d'équipements obsolètes tout en maintenant la configuration des équipements.

⁸ Un PSC dynamique est constitué d'une séquence de PSC déduite l'une de l'autre par l'ajout ou le retrait d'une contrainte [J&B 96].

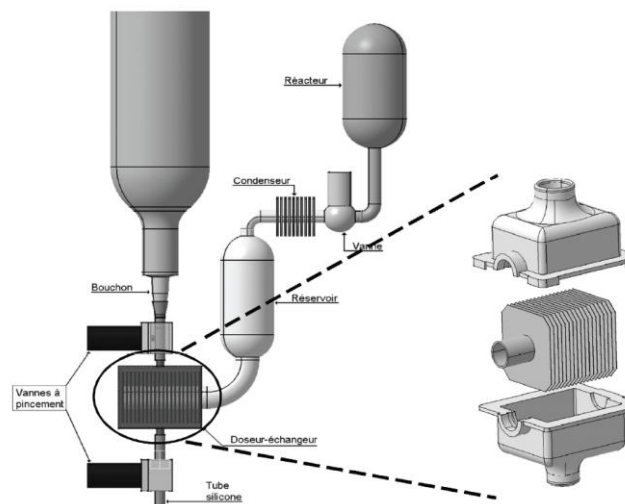
4 L'exemple d'application de PPC en conception

Etant donné que l'étude des systèmes liés à la conception représente le principal objectif de ce travail, un exemple de l'application des PSC est proposé afin d'illustrer et valider cette technique vis-à-vis des problèmes de conception de procédés. L'exemple a été pris parmi ceux proposés par [CHE 07]. Il porte sur la conception d'un échangeur thermique intégré à un doseur/refroidisseur d'apéritif.

4.1 Le problème de conception d'un échangeur thermique

Le problème décrit un système mécanique ayant pour objectif de refroidir une dose d'apéritif à 8°C en moins de 25s (figure 3.7). Ce système autonome ne comporte pas de compression frigorifique mais utilise, en revanche, une solution de sel pour ses facultés d'adsorption de l'ammoniac.

En fonctionnement, l'ouverture de la vanne provoque une chute de pression et l'ammoniac, se trouvant dans le tube du doseur-échangeur, se vaporise pour rejoindre l'adsorbent et y être adsorbé par la solution de sel. Cette évaporation entraîne une production de froid : c'est le principe de la réfrigération. Quand l'adsorbant est saturé, l'adsorption s'arrête ainsi que la production frigorifique. Il faut alors recharger le système en chauffant le sel. L'apport de chaleur induit la séparation du sel et de l'ammoniac gazeux qui rejoint alors son réservoir en se liquéfiant dans le condenseur. Dans l'exemple, la solution repose sur la partie liée au dosage/refroidissement de l'apéritif et il prend en compte les échanges de chaleur entre l'ammoniac et l'apéritif.



Source : [CHE 07]

Figure 3.7 Schéma du refroidisseur d'apéritif

Les flux fonctionnels correspondent aux transferts d'énergie calorifique entre l'apéritif et l'ammoniac. L'apéritif transfère de la chaleur aux ailettes de l'échangeur, qui la transfèrent ensuite au tube, qui finit par la transmettre à l'ammoniac.

Dans l'exemple, on suppose que le doseur-échangeur est calorifugé, et que les pertes d'énergie entre les corps 1 et 2 vers l'air extérieur sont négligeables. Une loi de conservation de l'énergie est appliquée. Pour ce faire, on étudie les transferts d'énergie au niveau des ailettes et on utilise une loi de comportement thermique, prenant en compte le gradient de température sur les ailettes selon leur épaisseur et leur matériau.

Nomenclature du problème :

A Surface d'échange (m²)

ail Référence du choix de la longueur des ailettes (-)

C_p Capacité calorifique de l'apéritif (J/(kg·K))

diam Référence du choix du diamètre du tube (-)

e Espace entre ailettes (mm)

f_i Coefficient de rendement d'ailette (-)

h Coefficient d'échange (W/ (m²·K))

mater Référence du choix des matériaux pour les ailettes (-)

N Nombre d'ailettes dans le doseur

Nu Nombre de Nusselt (-)

NUT Nombre d'unité de transfert (-)

t Temps de refroidissement (s)

T_f Température finale (K)

T_i Température initiale (K)

T_{sat} Température de saturation de l'ammoniac (K)

V Volume du doseur (l)

ε Efficacité du doseur (-)

η Rendement d'ailette (-)

λ Conductivité thermique des ailettes (W/m/K)

ρ Masse volumique de l'apéritif (kg/m³)

Dans le cadre du doseur-échangeur, la loi de conservation de l'énergie peut être exprimée grâce au nombre d'unité de transfert (NUT) du système :

$$NUT = \frac{h \cdot A \cdot \eta \cdot t}{V \cdot \rho \cdot C_p}$$

L'expression de h , doit prendre en compte l'hypothèse que le fluide est quasi-immobile entre les ailettes, du fait de leur proximité et de la viscosité de l'apéritif ($e < 4 * 10^{-3}$ m et $\mu > 2 * 10^{-3}$ kg/ (m · s)) :

$$h = Nu \frac{\lambda}{2e}$$

La connaissance du concepteur par rapport à ce type de phénomènes physiques est prise en compte pour déterminer la valeur du nombre de Nusselt de l'ordre de 4 pour effectuer les calculs. L'efficacité ε du système est calculée à l'aide de la relation :

$$\varepsilon = \frac{T_f - T_{sat}}{T_i - T_{sat}} \text{ et } \varepsilon = e^{NUT}$$

Dans l'exemple cinq variables de conception correspondent aux références du catalogue fournisseur, qui permettent ensuite d'exprimer A et λ comme des variables auxiliaires. Le nombre d'ailettes (N) et leur espacement sont des caractéristiques dimensionnelles et représentent de variables de conception. T_i , T_f , T_{sat} , sont toutes des variables auxiliaires, dont certaines sont fixées pour définir le contexte d'utilisation. Le modèle résultant comporte 14 contraintes, dont 6 qui décrivent un catalogue d'ailettes et de tubes (contraintes spécifiques), 4 non linéaires et 3 contraintes définies par morceaux.

4.2 Le modèle du PSC

Constantes :

Nom	Description
$dose = 0.04$	Volume de la dose à refroidir (l)
$V = 0.06$	Volume du doseur (l)
$eail = 0.0005$	Épaisseur des ailettes (m)
$T_i = 20$	Température initiale de l'apéritif (°C)
$T_f = 8$	Température finale de l'apéritif (°C)

Variables de conception :

Nom	Description
$mater : int = \{1, 2\}$	Type de matériau (-)
$diam : int = \{1, 2\}$	Diamètre de tube (-)
$ail : int = \{1, 2\}$	Longueur d'ailette (-)
$e : real = [0.001, 0.004]$	Espace entre les ailettes (m)
$N : int = [5, 20]$	Nombre d'ailettes (-)

Variables auxiliaires :

Nom	Description
$\lambda : int = [1, 200]$	Conductivité des ailettes (W/m/K)
$t : real = [11, 15]$	Temps de refroidissement (s)
$d : int = [0, 50]$	Diamètre du tube (mm)
$L : int = [0, 50]$	Côté de l'ailette (mm)
$Tsat : real = [-15, 2]$	Température de saturation (°C)

Variables alias :

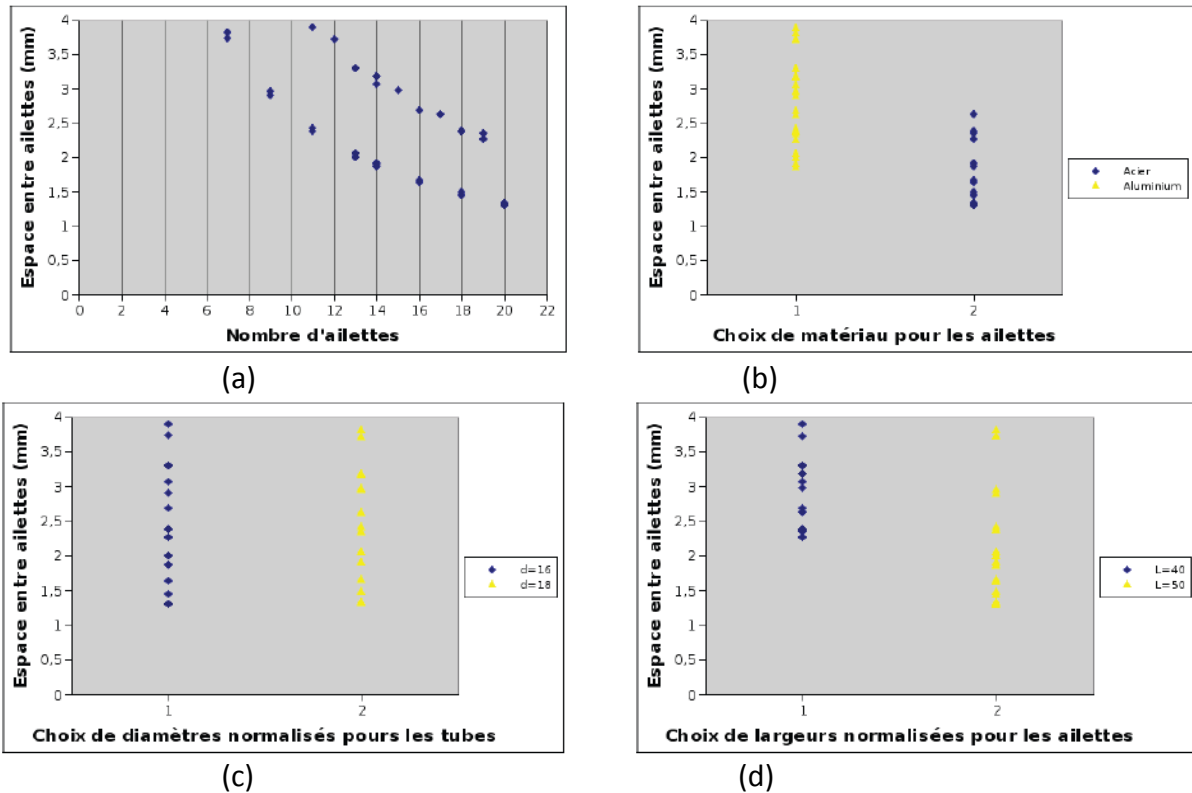
Nom	Contrainte	Description
A	$A = N \cdot (2 \cdot Aail + \pi \cdot e \cdot d) \cdot \frac{dose}{V}$	Surface de l'échangeur (m ²)
$Aail$	$Aail = \left(L^2 - \frac{\pi}{4} \cdot d^2 \right)$	Surface d'une demi-ailette (m ²)
fi	$fi = \frac{L - d}{2} \cdot \sqrt{\frac{2 \cdot h}{\lambda \cdot Aail}}$	Coefficient de rendement d'ailette (-)
η	$\eta = \frac{\varepsilon^2 \cdot fi - 1}{\varepsilon^2 \cdot fi + 1}$	Rendement d'ailette (-)
h	$h = \frac{1.2}{e}$	Coefficient d'échange du doseur (W/m ² /K)

Contraintes :

Relation	Description
$V = e \cdot Aail \cdot N$	Volume du doseur
$V \geq dose$	Condition de remplissage
$Tf = Tsat + (Ti - Tsat) \cdot e^{-\frac{h \cdot A \cdot \eta}{3.9 \cdot 10^6 \cdot dose}}$	Pour l'apéritif : $\rho \cdot Cp = 3.9 \cdot 10^6$
Piecewise (diam, [1, 1] : d = 16, [2, 2] : d = 18)	Diamètres normalisés pour les tubes
Piecewise (mater, [1, 1] : $\lambda = 200$, [2, 2] : $\lambda = 20$)	Matériaux des ailettes : 1 = acier, 2 = inox
Piecewise (ail, [1, 1] : L = 40, [2, 2] : L = 50)	Largeurs normalisées pour les ailettes

4.3 Les solutions

L'échangeur du doseur/refroidisseur d'apéritif est donc traité en utilisant les algorithmes appropriés (voir [CHE 07]) et 144 solutions sont trouvées. La figure 3.9 montre l'espace de solutions selon des perspectives différentes.



Source : [CHE 07]

Figure 3.8 Solutions en fonction : du nombre d'ailerettes (a), du matériau utilisé (b), du choix de diamètres de tubes (c) et de largeurs (d)

Sur la figure (a) l'espace de solution est divisé en deux. On identifie alors un premier groupe de solutions dont l'espacement entre ailettes ne descend pas en dessous de 2 mm et n'utilise que de 11 à 19 ailettes, alors que pour le deuxième groupe 7 à 20 ailettes sont utilisées et sont espacées de 1,2 à 4 mm. Les figures (b) et (c) laissent supposer que le choix du matériau et le diamètre du tube ne permettent pas d'identifier ces groupes, contrairement aux largeurs normalisées des ailettes dont la répartition correspond à celle de la figure (a). Le diamètre du tube ne semble pas déterminant dans la conception de cet échangeur, puisqu'autant de solutions sont trouvées pour les deux diamètres disponibles. Par contre, le choix de matériau est important, puisqu'avec l'aluminium les ailettes sont globalement plus espacées qu'avec l'acier.

5 Conclusion

Ce chapitre a permis d'introduire la notion et les bases des PSC ainsi que de présenter ce formalisme. Cette approche couvre un grand nombre de problèmes de divers domaines. Grâce à la façon de modéliser les problèmes, la recherche d'une solution est donc exploitée par les propriétés des contraintes qui réduisent l'espace de recherche grâce à un algorithme de résolution de PPC. Parmi ces principaux algorithmes de résolution des problèmes PSC, on a présenté les plus performants, à savoir les algorithmes basés sur le principe de retour arrière intelligent comme le Back Jumping. Ensuite, les techniques de consistance d'arc ont été aussi présentées. Ces techniques permettent d'améliorer considérablement l'exploration, en diminuant le nombre de nœuds à examiner.

La PPC offre aussi un cadre excellent pour l'application à des problèmes de conception car une fois les connaissances formalisées, une résolution indépendante du domaine d'application peut être réalisée. C'est cette formalisation qui permet une réutilisation des connaissances afin d'aider à la conception de futurs produits ou procédés. Grâce à l'avancée des recherches dans les méthodes de résolution de PSC, particulièrement dans les problèmes à caractère continu, les possibilités de résolution de ce type de problèmes ont été étendues. Ainsi par exemple, les problèmes mêlant des variables discrètes et continues, ce qui est souvent le cas en conception, peuvent être résolus par le biais de méthodes de PSC mixtes.

A noter que la conception est aussi un processus dynamique puisque des modifications (en fonctions des choix technologiques, évolutions de la demande...) peuvent être fréquentes, notamment lors de la conception préliminaire, où sont en jeu un grand nombre de décisions sur le système. Les modèles peuvent alors faire l'objet de plusieurs modifications. Le formalisme de la PPC facilite ces manipulations et permet d'envisager la possibilité de remémorer et exploiter plus facilement ces modèles (créés lors d'expériences antérieures).

A l'heure actuelle, les chercheurs se penchent de plus en plus sur les performances des traitements numériques (e.g. temps de calcul) mais la question se pose tout de même sur les possibles interactions avec l'utilisateur, car dans la plupart des méthodes de PPC, ce dernier n'exerce aucun contrôle durant le processus de résolution.

Finalement un exemple illustre l'efficacité de l'application des PSC dans le domaine de la conception de procédés. Une identification des différents types de variables et contraintes est aussi mise en avant pour clarifier la modélisation.

Le processus d'adaptation : fondements et problématique

Suite à l'étape de remémoration du RàPC, les cas sélectionnés sont employés pour dériver la solution au nouveau problème rencontré. Cette étape d'adaptation est primordiale pour concevoir un système de RàPC efficace (en termes de résolution de problème). Conceptuellement simple, cette étape est d'une concrétisation complexe. En effet, elle demande des connaissances supplémentaires et spécifiques qui génèrent des problèmes d'acquisition, formalisation, et de gestion. Dans ce chapitre on aborde cette problématique, d'abord en détaillant ses principes généraux, des enjeux et enfin l'approche retenue pour la suite de ce travail est présentée.

Chapitre 4 : Le processus d'adaptation : fondements et problématique

1 Contexte et principes sur l'adaptation en RàPC

De façon générale, l'adaptation en RàPC peut être définie, comme le fait de modifier la solution d'un cas remémoré afin qu'elle puisse répondre aux exigences du problème cible. Par exemple, le contexte d'un nouveau problème peut être défini une fois qu'un cas similaire a été remémoré, toutefois il n'en est pas de même avec la solution. L'adaptation devient donc l'art de concevoir une nouvelle solution. Dans ce cadre, l'identification de ce qu'on doit changer et comment ces modifications doivent être appliquées représentent les enjeux principaux.

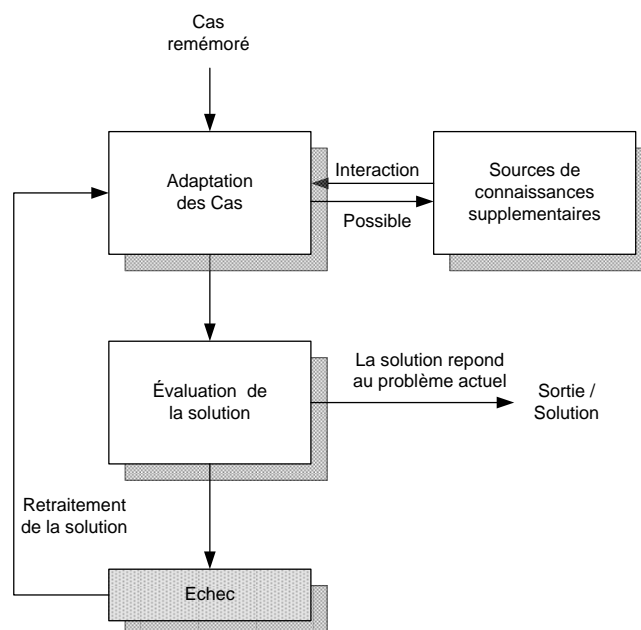
Dans la communauté du RàPC, l'utilité de l'étape d'adaptation a fait débat avec l'opposition de deux visions distinctes :

- D'un côté, plusieurs auteurs, comme l'explique [HAO 08] considèrent l'adaptation comme « *une étape délicate à mettre en place dans un système de RàPC en plus d'être spécifique au domaine d'application que l'on traite* ». C'est ainsi que certains auteurs évitent l'effort de recherche sur cette phase et préfèrent développer la partie remémoration [KAS 95] tablant sur l'abondance de cas et leurs capacités à couvrir l'espace des problèmes pour pallier la faiblesse de l'adaptation [S&W 86].
- De l'autre côté, des chercheurs qui basent leurs études sur cette étape primordiale en la considérant comme étant au cœur des systèmes de RàPC [LIE 04], [MIL 96]. [C&F 06] affirment même que « *l'adaptation confère au système de RàPC sa qualité de résolveur de problèmes* ».

Même si plusieurs 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 reste cruciale pour assurer l'efficacité d'un système de RàPC. Cependant elle reste une étape difficile à appréhender car elle demande des connaissances spécifiques. Ces dernières génèrent des problèmes d'acquisition, de formalisation et de gestion. Dans ce chapitre une étude sur l'adaptation et les différentes façons de la mettre en œuvre est proposée afin de mettre en relief son rôle et son importance, principalement pour les systèmes de RàPC appliqués à la conception.

1.1 Éléments fondamentaux sur l'adaptation

L'adaptation est un processus simple à comprendre qui consiste en la transformation de la solution source « *Sol (source)* » en une solution appropriée à cible : *Sol(cible)*. La figure 4.1 illustre l'étape d'adaptation comme étant au cœur de ce processus de transformation. On y aperçoit que l'implantation de ce processus demande l'ajout de connaissances supplémentaires pouvant provenir de diverses sources. L'adaptation en RàPC nécessite un effort important de réflexion et de savoir faire. En outre, l'adaptation peut être considérée comme l'étape la plus importante du RàPC puisqu'elle ajoute de l'intelligence à ce qui serait d'une certaine manière que des simples patrons ou tendances.



Source : adaptée de [P&S 04]

Figure 1.5 L'adaptation dans RàPC [S&S 04]

Figure 4.1 L'adaptation dans le RàPC

Pendant la phase d'adaptation, il est souhaitable de vérifier que la solution adaptée prenne en considération les différences entre le cas remémoré et le problème cible. Dans ce cadre, il y existe également une phase d'évaluation afin de valider la solution proposée. Si nécessaire, une itération supplémentaire peut être envisagée pour corriger ou ajuster les éléments de la solution proposée.

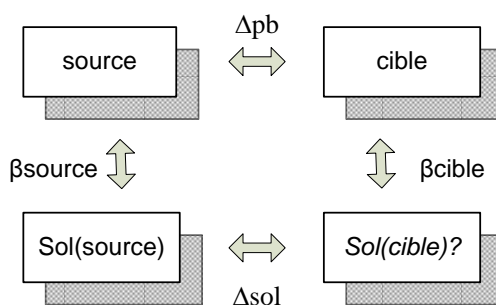
1.2 La formalisation de l'adaptation

Le formalisme qui explique le fonctionnement des premières étapes du cycle du RàPC, peut être exprimé par : trouver $(source, Sol(source)) \in Base\ de\ cas$ dont le cas « source » est

le plus semblable au cas « cible » selon un critère de similarité. Le but de l'adaptation est par conséquent de trouver la solution pour *cible* en modifiant $Sol(source)$. Il est donc possible de représenter l'adaptation grâce à la fonction suivante :

$$Adaptation : ((source, Sol(source)), cible) \rightarrow Sol(cible)$$

Chouraqui [CHO 86] décrit cette relation au travers d'une analogie : « ...en sachant que $Sol(cible)$ est à $Sol(source)$ ce que *cible* est à *source*, alors en connaissant la valeur de *source*, $Sol(source)$ et *cible*, il est possible de déterminer la valeur de $Sol(cible)$ ». La figure 4.2 traduit graphiquement le carré d'analogie énoncé dans la phrase précédente. Sur cette figure, les Δ sont les différences entre les cas *source* et *cible*. Ils matérialisent entre autres, les similarités et les différences entre ces deux cas. Δpb contient les informations sur les différences entre les problèmes *source* et *cible* (souvent nommé appariement entre problèmes). De façon similaire, ΔSol matérialise l'écart entre les solutions des cas *source* et *cible*. Une relation de dépendance β traduit le lien entre un problème et sa solution : *source* intervient dans la déduction de $Sol(source)$.



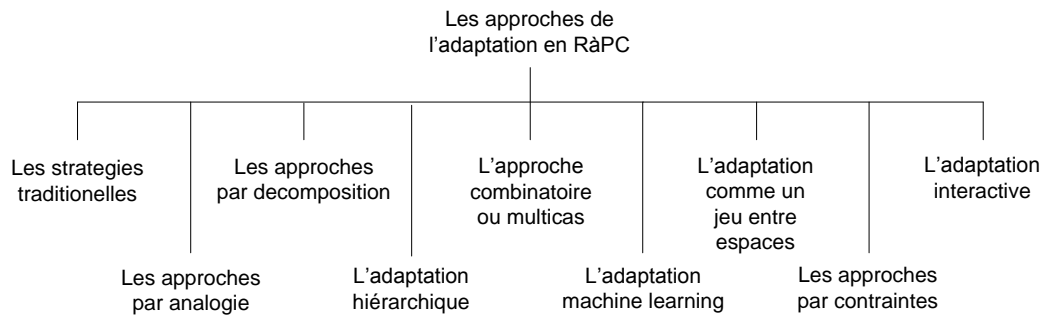
Source : [CHO 86]

Figure 4.2 Le carré d'analogie [CHO 86]

Dans un premier temps, il faut présenter les divers travaux et stratégies d'adaptation, ce que propose la section suivante. Ensuite une analyse est proposée dans la section 3, portant sur les différentes approches pour acquérir les connaissances d'adaptation. Finalement la stratégie d'adaptation proposée dans ce travail est discutée dans la section 4.

2 Les travaux sur l'adaptation : différentes approches

Plusieurs approches ont été proposées dans la littérature pour effectuer l'adaptation. En effet l'adaptation peut se servir de diverses techniques. Lieber dans [LIE 04] a réalisé un vaste état de l'art dans le but de recenser ces différentes stratégies d'adaptation et leurs différents domaines d'application. La figure 4.3 présente une taxonomie de ces différentes approches.



Source : adaptée de [BAD 09B]

Figure 4.3 Une taxonomie des différentes approches d'adaptation

Pour choisir une stratégie pour mener l'adaptation de cas, il peut être utile de se poser 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 ces interrogations, les travaux sur l'adaptation se déclinent principalement autour de trois points [LIE 04] :

- Les approches générales d'adaptation : qui proposent des modèles généraux pour effectuer l'adaptation, notamment sous forme de principes ou algorithmes.
- Les approches avec acquisition de connaissances supplémentaires, spécifiques à cette étape. Les connaissances utilisées pour effectuer l'adaptation sont désignées sous le nom de connaissances d'adaptation. Les démarches suivant ce principe, considèrent plutôt les moyens (principes généraux) d'explicitier l'adaptation dans le domaine étudié [HAO 08].
- Les catalogues de connaissances. A mi-chemin entre les deux approches précédentes, ils formalisent et utilisent des connaissances d'adaptation applicables à plusieurs domaines, des exemples sont donnés dans [KOL 93], [LIE 02] et une étude les recensant est faite dans [LIE 08].

2.1 Les stratégies traditionnelles d'adaptation

Une classification générale de l'adaptation proposée par [P&S 04] distingue trois stratégies majeures :

- *La reinstanciation* est la forme la plus simple d'adaptation, dans laquelle *Sol(source)* est simplement copiée à partir des cas recherchés et appliquée directement, sans modification. Dans [LIE 02] une étude est réalisée portant sur six types d'adaptation par copie : le raisonnement approximatif ou incertain, l'exploitation d'une équivalence entre représentation de problèmes, l'exploitation d'une relation de généralité entre problèmes, l'exploitation de symétries, le principe de précaution et l'exploitation de dépendances nulles.

- *La substitution* remplace les éléments de la solution qui sont inacceptables parce qu'ils entraînent un conflit ou contredisent les nouvelles conditions du problème.
- *La transformation* est utilisée lorsqu'aucun substitut n'est trouvé. Une solution sera alors dérivée en se basant sur les contraintes et les caractéristiques de la solution requise. Une contrainte spécifie quelles propriétés la solution devrait ou ne devrait pas avoir. La solution doit se conformer aux contraintes, et aucune contradiction ou conflit n'est permis. Pour identifier les contraintes, une certaine connaissance prédéfinie ou heuristique doit être disponible.

Il est à noter que ces stratégies sont complémentaires. En effet, en fonction du degré de similarité entre le cas source et le cas cible, on optera pour l'une ou l'autre des stratégies. Si la similarité entre les cas source et cible est très élevée et lorsque qu'aucune contrainte ou exigence n'est imposée sur la solution voulue alors la reinstanciation peut être utilisée, ou encore lorsqu'il s'agit d'un RàPC servant à justifier des choix. Dans cette stratégie la mesure de similarité et le choix du seuil sont les facteurs les plus importants. Lorsque *Sol(source)* est trop éloignée de l'objectif visé, alors certaines de ses parties sont modifiées en s'appuyant sur les différences entre les problèmes *cible* et *source*. Enfin, soit à cause d'exigences spécifiques, soit parce que le problème cible n'a pas « d'équivalent suffisamment proche » dans la base de cas, *Sol(source)* doit être partiellement voire totalement revue et transformée.

De par 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 en totalité. En fonction de la finalité du RàPC, on peut être amené à choisir une de ces trois stratégies (e.g. la reinstanciation pour un système d'aide à la décision). Lors de ce choix il faut également prendre en considération les coûts de l'implémentation d'une stratégie par rapport à une autre. Toutefois, dans un système de RàPC générique, il est recommandé de combiner ces stratégies pour mieux réussir l'étape d'adaptation

2.2 L'adaptation par analogie

En partant du principe résumé sur la figure 4.2 (section 1.2), l'adaptation peut être classifiée selon deux types d'analogie. A ce propos, Watson dans [WAT 97] les distingue de la façon suivante : l'adaptation transformationnelle dont le principe est d'appliquer des règles d'adaptation à *Sol(source)* et l'adaptation dérivationnelle où l'on réutilise les formules qui ont produit *Sol(source)* pour obtenir *Sol(cible)*. La figure 4.4 montre ces deux modes d'adaptation ainsi que leurs différentes étapes d'après [LIE 08].

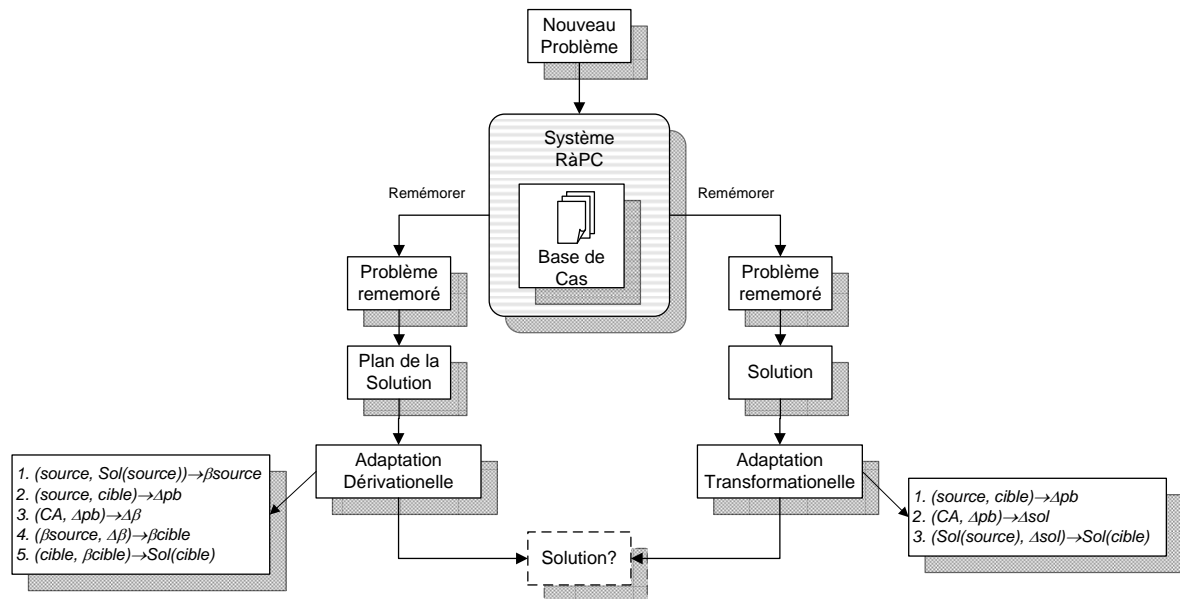


Figure 4.4 Adaptation dérivationnelle et transformationnelle

L'adaptation transformationnelle (aussi appelée « structurelle ») ne considère pas comment la solution a été obtenue. Dans ce cas de figure, les cas contiennent des descriptions et des solutions qui sont ensuite réutilisées pour apporter des modifications appropriées à la nouvelle situation [CAR 83]. La solution trouvée n'est pas une solution directe pour le nouveau problème, mais il existe une connaissance sous forme d'opérateurs transformationnels qui peuvent être appliqués à la solution remémorée. On distingue deux sous-classes dans l'adaptation transformationnelle :

- Par valeur : elle consiste à adapter les valeurs des attributs de la solution du cas source (lien avec la substitution de la section précédente).
- Par structure : elle consiste à adapter la structure de la solution du cas source (lien avec la transformation de la section précédente).

CHEF [HAM 89], est l'un des premiers systèmes RàPC mettant en application l'adaptation transformationnelle, ensuite il y en aura d'autres comme SPA [H&W 95] et DIAL [LEA 96A]. Cependant, ce qui distingue SPA d'autres systèmes transformationnels c'est son degré d'indépendance au niveau du domaine d'application, en effet les opérateurs transformationnels peuvent s'appliquer dans n'importe quel domaine particulier. PRIAR [K&H 92] est également un système de planification hiérarchique qui met en application des règles de transformation indépendantes du domaine.

L'adaptation dérivationnelle (ou générative) se focalise sur la méthode utilisée lors des épisodes antérieurs de résolution. Il s'agit d'exploiter les connaissances créées lors de la

démarche de résolution suivie pour construire de nouvelles solutions capables de s'adapter aux nouveaux problèmes. Dans ce type d'adaptation, on stocke pour chaque cas, la trace des étapes qui ont permis de générer sa solution (avec les informations nécessaires pour sa compréhension : décisions prises, connaissances utilisées...). Cette traçabilité est souvent utilisée en logique de conception.

L'adaptation dérivationnelle est souvent préférée à l'adaptation transformationnelle, parce qu'elle réduit l'espace de recherche en éliminant les choix infructueux. De plus elle offre plus de flexibilité puisque l'utilisateur peut réutiliser la trace dérivationnelle. Cette technique présuppose que les traces dérivatives existent. Lorsque ce n'est pas possible, l'analogie transformationnelle est donc la meilleure option [M&C 08].

Parmi les systèmes basées sur l'approche dérivationnelle on trouve notamment Prodigy [V&C 93] qui est un système de planification applicable dans plusieurs domaines, COBRA [SMY 93] pour la création de modèles physiques dans le domaine de l'analyse industrielle, DIAL [LEA 95] le système pour la planification d'interventions en réponse à des catastrophes d'origine naturelle ou humaine, IAKA-NF [COR 08] pour la résolution de fonctions numériques ou TAAABLE [BAD 09A] dans le domaine culinaire pour proposer à ses utilisateurs des recettes de cuisine.

2.3 Les approches par décomposition

Une autre stratégie d'adaptation consiste à fractionner un problème d'adaptation en sous-problèmes plus simples de façon à les résoudre séparément. Pour ce faire, trois différentes approches ont été proposées :

- Décomposition du problème d'adaptation. Une fois le problème d'adaptation décomposé, chaque problème est résolu indépendamment afin de combiner les solutions partielles en une seule solution *Sol(cible)*. On trouve ce type de stratégie notamment dans le système Déjà Vu [S&C 93].
- L'adaptation différentielle. Les variations entre les cas, à partir d'un ensemble de dépendances entre descripteurs de problèmes et de solutions, sont modélisées localement grâce à une formule basée sur le calcul différentiel. Elle lie ces variations (exprimées comme des différentielles en un point (x, y) donné) aux dérivées partielles pour calculer *Sol(cible)*. Cette démarche est utilisée dans l'algorithme IAKA-NF [COR 08].
- L'adaptation par reformulations. Elle combine l'adaptation par décomposition et la recherche dans un espace d'états. L'application de reformulations (chacune représentant

un ensemble de liens entre l'espace de remémorations et l'espace des adaptations) permet de décomposer la tâche d'adaptation en introduisant des problèmes virtuels. Le système TAAABLE [BAD 09A] en est un exemple.

2.4 L'adaptation hiérarchique

Dans [B&D 06], une explication a été donnée à propos de ce type d'adaptation : « ...les cas sont stockés à plusieurs niveaux d'abstraction. L'adaptation est effectuée du plus haut niveau vers le plus bas niveau. Au début, la solution est adaptée au niveau d'abstraction le plus élevé (les détails moins appropriés sont omis). Puis, la solution est affinée par étapes et les détails exigés sont ajoutés. Pour les différents niveaux d'abstraction, l'adaptation hiérarchique peut réutiliser des cas simples ou des cas multiples en utilisant différents détails ou différentes parties des solutions réutilisées ».

On trouve ce type d'adaptation dans un certain nombre de systèmes, notamment dans Déjà Vu [S&C 93], où les objets sont décrits dans une hiérarchie de composition, et l'adaptation est effectuée à différents niveaux de granularité. Dans [LOP 05], le système génère des modèles de différents niveaux d'abstractions à partir de descriptions détaillées de processus afin de réduire la complexité du traitement. Finalement dans SAARA [B&D 06] l'adaptation hiérarchique intervient lors de la construction de la solution associée au cas cible.

2.5 L'adaptation combinatoire ou multi-cas

Dans l'adaptation combinatoire, il s'agit de remémorer plusieurs cas *source* afin de composer *Sol(cible)* à partir des différentes *Sol(source)* remémorées. [W&B 98] a nommé ce type d'adaptation comme l'adaptation compositionnelle. Elle peut être utilisée dans deux situations [A&B 00] :

- Lorsque la solution se compose de différentes parties indépendantes, les composants peuvent être adaptés avec plus ou moins de précision. Cette méthode est efficace s'il y a peu de conflits entre les composants [W&B 98].
- Lorsque la solution ne peut pas être divisée en parties indépendantes mais que les solutions des cas similaires doivent être combinées.

Plusieurs systèmes utilisent cette stratégie : CADSYN [M&Z 93], Prodigy/Analogy [VEL 94], PRODIGY [H&V 95], CAPLAN/CBC [M&W 96], EADOCS [N&V 96] et COMPOSER [P&P 97].

2.6 L'adaptation « Machine Learning »

Récemment, de nombreuses approches de « machine apprenante » ont été développées pour résoudre le problème d'adaptation. Dans ce type d'approche, l'apprentissage produit des heuristiques spécialisées qui se basent sur les différences entre *cible* et *source*. Ces heuristiques peuvent être employées pour déterminer la quantité d'adaptation qui est exigée. Un des avantages de ces techniques est que la connaissance d'adaptation peut être acquise automatiquement à partir de la base de cas. De plus, la maintenance de cette connaissance peut être plus facilement gérée et contrôlée en l'actualisant et en la vérifiant lors de confrontation avec des nouveaux cas. Elles utilisent plusieurs techniques de calcul dont par exemple :

- Les réseaux bayésiens. Les différences entre *cible* et *source* sont soumises à un modèle bayésien. La sortie identifie la stratégie d'ajustement, comme par exemple le modèle dans les travaux de [M&T 94]. Leur modèle propose une interprétation bayésienne de l'espace entre les cas permettant de pallier l'appariement (matching) des cas et leur adaptation grâce à la propagation de la probabilité.
- Les réseaux de neurones. Corchado et Lees [C&L 01] utilisent des réseaux de neurones artificiels de type « Radial Basis Function ». L'apprentissage du réseau est réalisé par un ensemble de cas *source*, sélectionnés lors de la phase de remémoration. Bouabdallah et Chikh [B&C 07] proposent une technique qui décompose la base de cas en groupes de similarité. Chaque groupe rassemble un ensemble de cas qui partagent un même sous ensemble d'attributs. Puis la technique associe à chaque groupe un réseau de neurones qui sera entraîné en utilisant les cas de ce groupe. Ensuite il présente *cible* en entrée du réseau de neurones et *Sol(cible)* est obtenue après un calcul effectué par ce dernier.
- Les algorithmes génétiques. Dans les deux techniques précédentes, la connaissance d'adaptation est extraite directement à partir des cas. Toutefois, lorsque la base de cas n'est pas suffisamment abondante, l'emploi d'algorithmes génétiques peut être envisagé. L'idée est que quand une solution est remémorée, un processus de modification est initialisé aléatoirement. La solution adaptée est examinée et une rétroalimention est conduite pour déterminer les ajustements. Ce processus est répété jusqu'à ce qu'une solution pertinente soit calculée. Un exemple est présenté dans le système GENCAD [S&M 99] dans le domaine de la configuration. Les cas sélectionnés lors de la phase de

remémoration forment la population initiale pour les algorithmes génétiques. Deux types d'adaptation sont utilisés pour faire évoluer le contenu des cas remémorés jusqu'à obtenir une solution satisfaisante :

1. structurelle : par combinaison ou croisement,
2. paramétrique : par mutation.

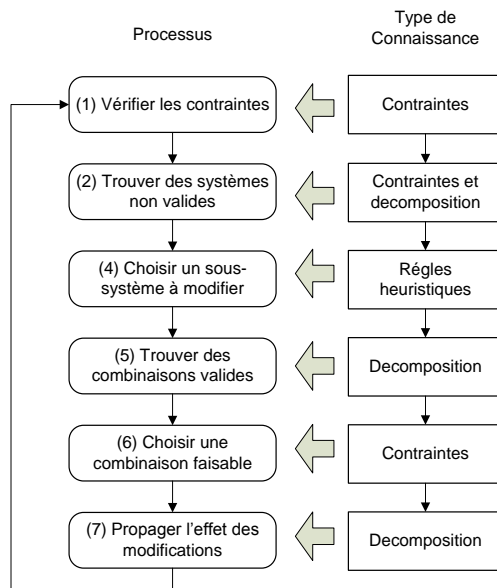
2.7 L'adaptation comme un jeu entre espaces

Elle est formulée à partir d'un jeu entre deux espaces : les remémorations et les adaptations. D'un côté, l'ensemble des correspondances entre *cible* et *source* définit l'espace des remémorations et de l'autre l'espace des adaptations désigne un ensemble de fonctions qui peuvent être utilisées pour modifier $Sol(source)$ afin de trouver $Sol(cible)$. [BAD 09B] identifie, dans cette approche, les connaissances d'adaptation comme étant celles qui permettent de lier ces deux types d'espaces. Un exemple est proposé dans le système Déjà Vu [S&C 93].

2.8 L'adaptation par contraintes

L'adaptation peut être résolue en l'abordant comme un PSC. Cette adaptation essaye d'éliminer les violations des contraintes dans une solution potentielle en recherchant les substitutions acceptables pour $Sol(cible)$. Le processus d'adaptation basé sur une approche de contraintes se caractérise comme la recherche des affectations consistantes pour *cible* à l'aide de connaissances disponibles. Ces dernières pouvant être extraites directement de la base ou auprès de l'expert.

Dans [MAH 95] un modèle général sur le processus d'adaptation en employant les PSC est proposé (figure 4.5). Les cas sont d'abord décomposés en sous-systèmes dans la base de cas. Une fois *source* remémoré, il commence par vérifier les contraintes violées par *cible*. Puis il cherche et sélectionne un sous-système pour le modifier et génère toutes les combinaisons possibles pour ce sous-système. Ensuite il sélectionne une combinaison de valeurs comme une nouvelle description du sous-système et finalement il propage l'effet des modifications. Des systèmes comme JULIA [HIN 92], CADRE [HUA 96], ou COMPOSER [P&P 97] ont prouvé l'efficacité de l'approche par contraintes particulièrement appliquée dans le domaine de la conception. Une étude plus détaillée sur ces travaux sera présentée dans le chapitre 5.



Source : [MAH 95]

Figure 4.5 Le principe du processus d'adaptation en utilisant les PSC

2.9 L'adaptation interactive

Ce type d'adaptation se distingue des autres en ce qu'il se déroule grâce à l'intervention ponctuelle de l'utilisateur. D'ailleurs, dans les premières années d'utilisation du RàPC, une partie des chercheurs [LEA 96], considéraient l'adaptation automatique comme étant essentielle pour attribuer aux systèmes un haut degré d'autonomie et par conséquent une meilleure capacité à résoudre le problème d'adaptation en émulant l'action d'un expert. Cependant, d'autres auteurs, comme [MAR 96], contredisaient cette idée en soutenant que l'automatisation dans l'adaptation n'était ni faisable pour arriver à une adaptation réussie ni indispensable pour assurer l'efficacité d'un système de RàPC. Même si des techniques ont été récemment développées dans ce sens (e.g. machine apprenante, techniques d'exploration de données...), l'automatisation se révèle bien faisable avec des résultats satisfaisants. Toutefois un problème est rencontré car l'efficacité et la performance de ces méthodes dépendent considérablement de l'abondance de cas dans la base.

En revanche, de nombreux systèmes se basant sur une adaptation interactive, se sont montrés plus avantageux, même lorsqu'il s'agit d'adapter des problèmes complexes. Cette intervention de l'utilisateur peut être réalisée de diverses manières [BAD 09B] :

- Une adaptation manuelle, comme dans le système DIAL [LEA 96A].
- Par le biais d'un modèle de préférences de l'utilisateur comme le système WebAdapt [L&P 07].

- A travers un processus itératif : le système propose une solution que l'utilisateur révisé et modifie selon ses besoins. Ses connaissances peuvent éventuellement être acquises, et réutilisées dans l'avenir (acquisition de connaissances d'adaptation auprès des experts). Les systèmes FrakaS et IakA [COR 08] sont des exemples de ce type d'adaptation.

3 L'acquisition des connaissances d'adaptation

Pour être performante et efficace, la phase d'adaptation nécessite l'ajout de nouvelles connaissances, appelées connaissances d'adaptation (CA). Au cours de ces dernières années, leur acquisition a été étudiée par divers auteurs et sous différents angles [HAN 95], [P&S 04], [LIE 04], [C&F 06], [BAD 09A]. Pour acquérir ces connaissances, quatre principales rubriques sont identifiées (figure 4.6) : la source d'où sont extraites les connaissances, la forme sous laquelle elles sont acquises, la méthode utilisée pour leur acquisition et finalement le moment où cette acquisition est réalisée.

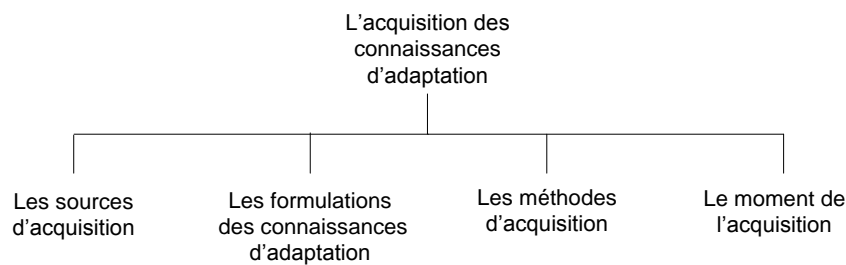


Figure 4.6 Une taxonomie des différentes approches pour l'acquisition de connaissances d'adaptation

Afin d'éviter des confusions lors de l'utilisation des concepts pour expliquer les interactions humaines, il est nécessaire de distinguer trois types principaux d'acteurs :

- *L'expert*. C'est l'expert du domaine technique. C'est lui qui possède les connaissances liées au domaine d'application.
- *L'utilisateur*. C'est un utilisateur averti du système qui possède un certain niveau d'expertise sur le domaine d'application. Cela peut lui permettre de prendre des décisions sans avoir besoin d'une validation de la part de l'expert du domaine.
- *L'usager (non expert)*. C'est l'utilisateur dépourvu de connaissances sur le domaine technique. Son intervention dans le système se limite à l'utilisation du système de RàPC et son interaction avec celui-ci doit être supervisée par un expert.

3.1 Les sources des connaissances d'adaptation

Les connaissances d'adaptation peuvent être extraites à partir de diverses sources. [HAN 97] identifie deux sources principales : la base de cas et l'expert du domaine. Plus tard [LIE 04] considère les critères d'évaluation comme étant une autre source de connaissances. [BAD 09B], quant à lui, décline quatre sources principales : les connaissances du domaine, le Web, la base de cas et l'expert et/ou utilisateur. Reprenant la vision de [WIL 97] sur une classification plus générale, deux grandes sources sont identifiées : les sources internes et externes au système de RàPC.

3.1.1 Les sources internes

D'après [WIL 97] ces sources exploitent les connaissances qui se trouvent à l'intérieur même du système de RàPC pour inférer de nouvelles connaissances. Deux types de sources internes sont usuellement utilisés : la base de cas et les connaissances du domaine.

- Dans le premier groupe, les connaissances d'adaptation sont extraites à partir des différences entre les cas présents dans la base de cas [S&K 96] et [JAR 01]. Selon [H&K 97], l'idée principale est que ces différences sont représentatives des différences qui pourraient être rencontrées entre *cible* et *source* lors d'un futur épisode. Des exemples de systèmes dont la source principale des connaissances d'adaptation est la base de cas : PRODIGY [V&C 93], PADIM [FUC 95], EADOCS [N&V 96] et KASIMIR [AQU 04].
- L'idée principale du deuxième groupe, selon [BAD 09B] est que le système possède suffisamment de connaissances du domaine pour faire face à tout nouveau problème d'adaptation. L'approche s'applique bien lorsque le système utilise des connaissances du domaine pour représenter les cas. Elles peuvent alors être réutilisées pour produire des connaissances d'adaptation. Les systèmes DIAL [LEA 96A], CHEF [HAM 89] ou CASEY [KOT 88] utilisent des connaissances du domaine comme sources de connaissances d'adaptation.

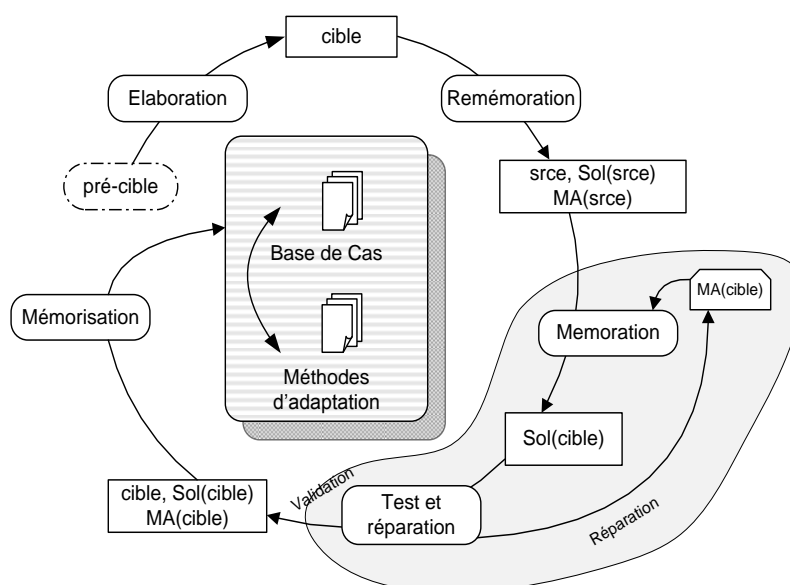
Les sources internes sont censées être à moindre coût et en général elles tiennent leur place après la phase de résolution des problèmes. L'avantage principal est que les techniques d'acquisition se basant sur des sources internes de connaissances sont facilement operationalisables. [C&F 06] observent quelques inconvénients pour les systèmes de RàPC à sources internes : l'inférence de nouvelles connaissances est limitée au vocabulaire des cas présents dans la base. Elles ne permettent pas d'inférer des connaissances qui ne sont pas

explicables avec les connaissances présentes dans le système et enfin elles réduisent l'intervention de l'utilisateur.

3.1.2 Les sources externes

Elles proviennent de l'extérieur du système de RàPC et peuvent communiquer avec celui-ci de plusieurs manières : interfaces, base de données, réseaux... En général elles sont activées pendant le processus de résolution [C&F 06]. Deux principales sources externes sont identifiées : l'expert ou l'utilisateur et le web.

- Un expert du domaine (ou un utilisateur) qui est sollicité pour formuler les connaissances nécessaires pour le processus d'adaptation. Cette communication peut s'établir grâce à une interface homme-machine qui permet une telle acquisition. Dans les travaux de [COR 07A] une modification du cycle de RàPC est faite en ajoutant une boucle d'interaction avec l'utilisateur (figure 4.7).
- Dernièrement le web offre de nombreuses possibilités grâce à la multitude des bases de connaissances disponibles. Le système WebAdapt [L&P 07] exploite les connaissances disponibles sur le web pour acquérir les connaissances d'adaptation. Il existe aussi une direction de recherche qui consiste à s'appuyer sur les langages du Web Sémantique comme par exemple le système CREEK [FID 06] qui intègre un vocabulaire spécifique, ou par le biais d'ontologies comme CBROnto [D&G 02].



Source : [COR 07A]

Figure 4.7 La boucle d'interaction avec l'utilisateur

Les systèmes exploitant de telles sources ont l'avantage d'être plus flexibles lors de la résolution puisqu'ils ne se limitent pas aux connaissances présentes à l'intérieur du système mais ils étendent leurs possibilités en incorporant de nouvelles sources.

3.2 Les formulations des connaissances d'adaptation

Que ce soit automatiquement ou grâce à l'intervention d'un expert, il faut trouver un moyen de coder ces connaissances dans le système de RàPC. 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 des experts). [C&F 06] effectuent une analyse plus étendue où ils classifient les différentes formes des connaissances d'adaptation en quatre groupes : les règles d'adaptation, les opérateurs d'adaptation, les reformulations et les cas d'adaptation.

3.2.1 Les règles

L'idée est d'extraire à partir de la base de cas ou auprès d'un expert, des règles d'adaptation. En effet, la base de cas peut comporter une richesse en termes de connaissances implicites du domaine et il est logique d'essayer d'utiliser ces connaissances dans l'apprentissage des règles d'adaptation. Les règles ainsi construites peuvent être affinées et généralisées. Les expériences des experts peuvent aussi être formalisées comme des règles, souvent elles représentent les explications ou les décisions prises durant l'adaptation. Plus formellement, les règles prennent la forme suivante :

Si *Condition(Source, Sol (Source), Cible)*
alors *Modification(Source, Sol (Source), Cible)→Sol(Cible)*

Le système Déjà Vu [S&C 93] utilise des règles simples d'adaptation pour résoudre les différences entre les descripteurs de *cible* et de *source*. Dans [H&K 96] des mesures de confiance calculées en fonction du degré de généralisation peuvent être associées à chaque règle. On trouve l'utilisation des règles aussi dans [B&M 01] pour l'adaptation de problèmes de formulation chimique. Dans le système MIKAS [K&H 01] les règles sont obtenues auprès d'un expert, l'approche est utilisée dans le domaine de créations de menus selon des exigences nutritionnelles. Avec ce type de formalisation, on retrouve les mêmes problèmes que les systèmes experts, à savoir la gestion et la maintenance des règles produites.

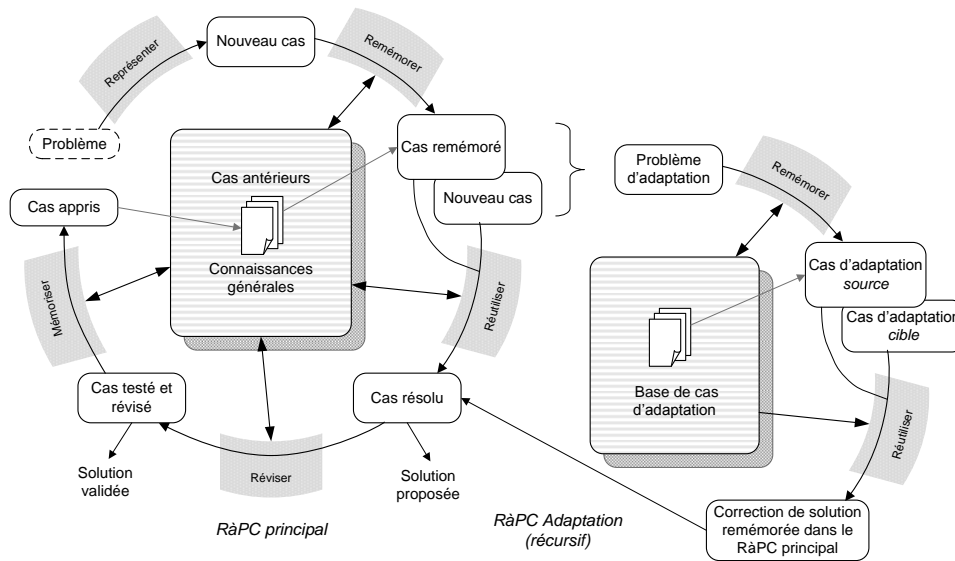
3.2.2 Les opérateurs d'adaptation

Les premiers travaux portant sur l'étude des connaissances d'adaptation, remontent à ceux de [HAN 95] où quatre types d'opérateurs d'adaptation sont identifiés :

- Les opérateurs d'élaboration de cible qui élaborent une description de *cible* pour faciliter l'adaptation. Cette élaboration ajoute de nouvelles informations ou redéfinit des parties de *cible* ; e.g. CYCLOPS [NAV 88] ;
- Les opérateurs de substitution des rôles qui effectuent des substitutions des différentes parties d'une solution (actions, valeurs ou rôles) à plusieurs niveaux de granularité, comme dans PERSUADER [SYC 87] et le système CHEF [HAM 89] ;
- Les opérateurs de sous-cibles qui décomposent la tâche d'adaptation en plusieurs parties plus simples, si la solution nécessite diverses adaptations, les opérateurs peuvent séquencer ces adaptations comme dans Déjà Vu [S&C 93] qui délègue la tâche d'adaptation à des spécialistes, chaque spécialiste apporte donc une modification locale ;
- Les opérateurs d'interaction selon le but ; cette interaction est réalisée entre les parties qui composent les solutions. Ils gèrent la détection et la réparation de mauvaises interactions qui peuvent se produire entre ces parties. Dans Déjà Vu [S&C 93], des stratégies générales d'adaptation sont utilisées pour manipuler de mauvaises interactions produites à partir de changements locaux faits par les spécialistes chargés de l'adaptation.

3.2.3 Les cas d'adaptation

Ces types de connaissances d'adaptation s'inscrivent dans l'approche d'adaptation par analogie dérivationnelle (point 2.2). Les traces des actions qui ont lieu durant la phase d'adaptation peuvent être stockées comme des cas d'adaptation [C&F 06]. Pour les cas d'adaptation, l'adaptation est elle-même effectuée avec un processus de RàPC, selon le principe nommé RàPC récursif [S&B 00] car le système conçu intègre deux processus de RàPC (figure 4.8). De nombreux auteurs étudient ce mode d'adaptation car il est facilement automatisable (les approches diffèrent dans la construction et l'utilisation des cas d'adaptation). Les cas d'adaptation (stockés dans une base spécifique ou créés à partir de la base de cas) se composent de (*source*, *Sol(source)*, *cible*)^A pour décrire le problème d'adaptation et de *Sol(cible)*^A pour sa solution.



Source : [S&B 00]

Figure 4.8 Principe du RàPC récursif

Lors de la soumission d'un nouveau problème au système, le premier processus de RàPC permet de rechercher le cas le plus similaire dans la base de cas. Le cas source remémoré et le problème cible initial forment ainsi un cas d'adaptation, puis le second système de RàPC est activé pour remémorer dans la base de cas d'adaptation le ou les cas d'adaptation le(s) plus similaire(s). Ce (ou ces) cas d'adaptation est (sont) ensuite utilisé(s) pour corriger la solution du premier cas source remémoré et ainsi proposer une solution adaptée. [JAR 01] ont mis en œuvre cette technique pour la conception de médicaments : choix des produits actifs, excipients... et de leurs dosages.

3.3 Les méthodes utilisées pour acquérir les connaissances d'adaptation

Dans [BAD 09B], quatre types de méthodes pour l'acquisition de connaissances d'adaptation sont identifiés. Ils se différencient par la façon dont l'acquisition est réalisée : manuelle, automatique, semi-automatique, ou interactive.

3.3.1 Les approches manuelles

Les connaissances d'adaptation sont acquises par le biais d'interviews d'experts durant lesquelles un transfert d'expertise est réalisé. Pendant ces sessions d'acquisition de connaissances, effectuées hors d'un épisode de résolution, l'expert est mis en situation et est encouragé à formuler ses connaissances qu'il utilise pour résoudre les problèmes. Les connaissances ainsi acquises peuvent être alors formalisées sous différentes formes (e.g.

règles, cas d'adaptation). Acquérir la connaissance d'adaptation auprès d'experts métier demande 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. De nombreux systèmes utilisent cette approche : CHEF [HAM 89], CLAVIER [H&H 92] et JULIA [HIN 92].

3.3.2 Les approches automatiques

Elles consistent à appliquer un algorithme d'apprentissage automatique pour obtenir des connaissances d'adaptation à partir d'un ensemble de données ou de connaissances. L'expert du domaine ne fait pas partie du processus. Elles 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). Dans cette catégorie on retrouve principalement les approches qui utilisent des techniques de « machine apprenante ».

3.3.3 Les approches semi-automatiques

Elles comprennent des interactions avec deux types d'acteurs : l'expert et l'utilisateur. D'après [BAD 09B], dans les approches semi-automatiques avec une interaction de l'expert, des éléments de connaissance sont générés grâce à des techniques d'extraction de connaissances à partir de données. Ces éléments sont ensuite présentés, hors de l'épisode de résolution, à un expert du domaine pour validation.

Lors de l'interaction avec d'utilisateurs, l'acquisition de connaissances d'adaptation est réalisée par un algorithme d'apprentissage déclenché durant le processus d'adaptation. L'apprentissage est guidé par le problème à résoudre et son paramétrage résulte d'un ensemble d'interactions avec l'utilisateur. Ce dernier intervient également pour valider les connaissances acquises. Un exemple des deux approches se trouve dans les systèmes CABAMAKA [AQU 07] et TAAABLE [BAD 09A].

3.3.4 Les approches interactives

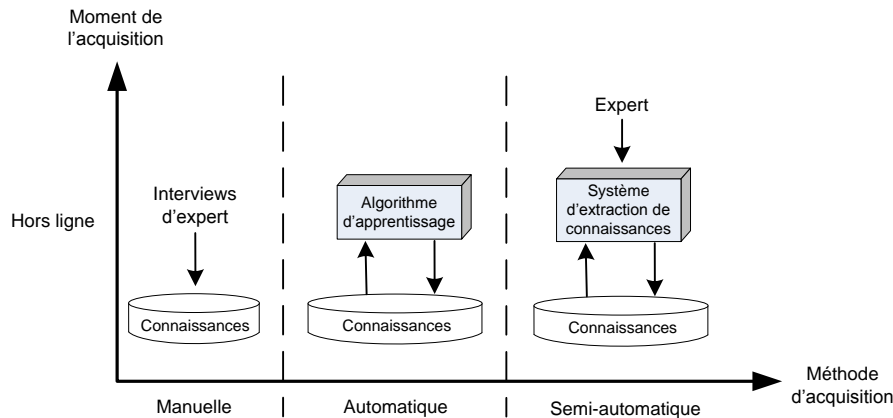
Il s'agit de tirer profit des interactions de l'utilisateur avec le système lorsqu'un échec d'adaptation se produit et qu'il est nécessaire de réparer la solution. Cette extraction de nouvelles connaissances peut se révéler moins consommatrice en temps que les approches manuelles mais plus coûteuse que les approches automatiques principalement à cause du nombre d'itérations nécessaires pour acquérir les connaissances appropriées. Cependant, les connaissances acquises auprès de l'utilisateur possèdent l'avantage d'être faciles à comprendre car elles sont souvent accompagnées d'explications. Cette approche se présente sous le nom d'acquisition opportuniste des connaissances dans les travaux de [COR 07A]. La difficulté principale est de mettre en place des interactions avec l'utilisateur de la façon la moins contraignante possible tout en essayant d'obtenir suffisamment d'informations.

3.4 Le moment de l'acquisition

Dans [COR 07A] deux moments sont identifiés pour acquérir les connaissances d'adaptation : en ligne ou hors ligne. [BAD 09B] les met en relation avec les méthodes d'acquisition et les schématise sur les figures 4.9 et 4.10.

3.4.1 L'acquisition « Hors ligne »

Les stratégies opérant hors-ligne acquièrent les connaissances d'adaptation en dehors de la phase de raisonnement. L'expert du domaine par exemple, peut être sollicité en dehors de l'épisode de résolution pour formuler les connaissances à réutiliser à l'avenir (méthode manuelle). Cela peut assurer la qualité dans les connaissances acquises mais s'avérer très pénalisante en temps. Quant à l'acquisition semi-automatique, elle peut demander un effort d'ingénierie important de la part de l'expert car ce dernier doit, durant une phase de maintenance, traiter un nombre important de connaissances générées par un processus d'apprentissage automatique [COR 07A]. En général, ces stratégies sont assez dépendantes d'un processus externe pour acquérir les connaissances et les maintenir. Elles possèdent l'avantage de traiter un grand nombre de connaissances mais en demandant un effort plus important.

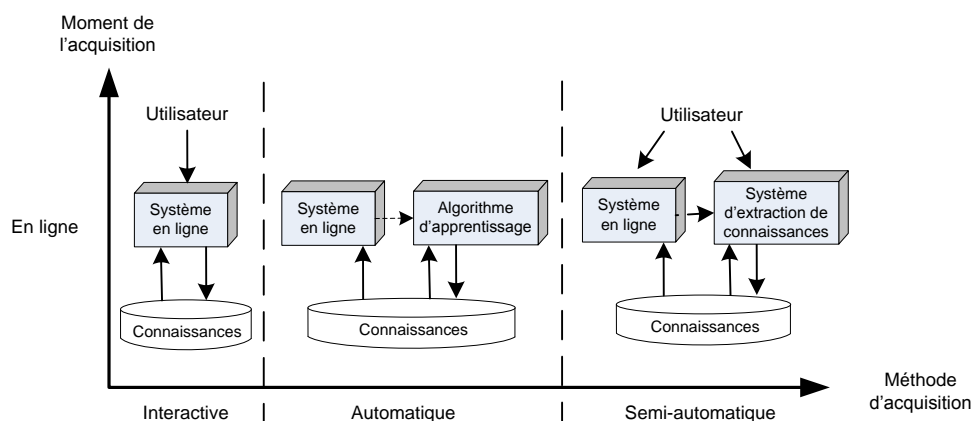


Source : [BAD 09B]

Figure 4.9 L'acquisition hors ligne

3.4.2 L'acquisition « En ligne »

L'acquisition des connaissances en ligne, consiste à profiter de l'étape de révision du RàPC pour solliciter ponctuellement l'utilisateur (approches interactive et semi-automatique) ou déclencher un algorithme d'apprentissage (approche automatique), afin d'acquérir les connaissances d'adaptation. Ces approches d'acquisition des connaissances sont motivées par la volonté de réduire l'effort d'ingénierie de la connaissance en sollicitant l'expert de manière ponctuelle lorsqu'il utilise le système pour résoudre un problème [COR 07A]. Toutefois, un inconvénient majeur peut être le nombre d'itérations que peut nécessiter le processus d'adaptation si l'utilisateur produit un grand nombre de connaissances non cohérentes et sans la supervision d'un expert.



Source : [BAD 09B]

Figure 4.10 L'acquisition en ligne

4 Positionnement de l'approche proposée d'adaptation

Les sections précédentes ont mis en évidence l'importance de la phase d'adaptation et les différentes approches pour l'aborder. Il faut noter que lorsqu'on utilise le RàPC comme un outil pour aider la conception, l'adaptation de cas représente un processus complexe pour suggérer des solutions cohérentes au problème initial. Cette complexité est intrinsèquement liée aux problèmes de conception et héritée à l'activité d'adaptation. En effet, elle peut s'avérer beaucoup plus complexe que le simple fait d'apporter des modifications à un cas de conception précédemment résolu [MAH 95].

L'adaptation de cas en conception représente tout un processus en lui-même. Un mécanisme efficace d'adaptation ne se limite pas à l'identification des différences entre le contexte des problèmes *source* et *cible* et la modification de *source* en prenant compte ces différences, il doit répondre à d'autres questions portant sur la représentation des connaissances, le maintien des modifications cohérentes ou la vérification des solutions. Tous ces aspects sont à prendre en considération lors de la mise en place d'une stratégie d'adaptation.

Les méthodes traditionnelles sont souvent insuffisantes pour répondre à ces dernières questions et surtout pour satisfaire tous les besoins de conception. Par exemple, les méthodes se basant sur la substitution ne prennent pas compte l'interaction entre les différentes parties. Par ailleurs, en utilisant des méthodes trop généralistes, le problème de conception est mené par des approches indépendantes du domaine sans prendre en compte les spécificités du processus de conception. Dans ce cadre, l'approche de satisfaction de contraintes semble être pertinente pour faciliter l'adaptation de cas en conception. Les travaux de [VAR 11] soulignant l'avantage de combiner cette approche qui utilise une connaissance formalisée (e.g. PSC) avec la connaissance contextualisée gérée par le système de RàPC.

Les démarches d'acquisition de connaissances d'adaptation ont l'avantage de rendre le RàPC très opérationnel et plus efficace par rapport aux systèmes de RàPC utilisant des méthodes traditionnelles qui ne prennent pas en compte ces connaissances. Ces démarches considèrent à l'adaptation dans le cadre d'un domaine d'application ayant pour but de mettre en évidence les principes ou moyens destinés à la clarifier. La plupart des auteurs s'accordent sur le fait que l'ajout de connaissances du domaine facilite énormément l'adaptation. Des applications dans le domaine de la conception ont déjà prouvé ces avantages. Pour ce travail,

trois différentes stratégies ont été d'abord choisies, parmi celles déjà présentées, et sont par la suite analysées afin de proposer une approche pour mener l'adaptation de cas en RàPC visant une application dans le domaine de la conception des procédés.

La première alternative envisagée consiste à tirer profit de la base de cas en y extrayant les connaissances d'adaptation grâce à des techniques de « machine apprenante » ou des techniques d'extraction de connaissances dans des bases de données (e.g. exploration de données). Ces techniques automatiques permettent de générer des heuristiques spécifiques qui permettent de prendre en considération les différences entre les spécifications d'entrée (i.e. attributs de problème) et les spécifications de sortie (i.e. attributs de solution). Ces heuristiques peuvent être alors employées pour déterminer la quantité d'adaptation qui convient [P&S 04]. Toutefois elles nécessitent d'avoir une base de cas contenant de nombreux cas afin d'éviter les imprécisions et approximations sur les heuristiques. Malheureusement, en conception, il est rare de posséder une base de cas suffisamment fournie pour ces techniques. Par ailleurs, l'acquisition automatique de ces connaissances ne permet pas toujours de générer des connaissances très pertinentes et facilement réutilisables par rapport à l'objectif fixé, de plus l'expert du domaine ne fait pas partie du processus.

Une deuxième alternative porte sur la capitalisation des connaissances d'un expert du domaine de façon manuelle. Celle-ci est sûrement l'approche la plus adaptée lorsque le RàPC est dédié à la conception de systèmes. Malheureusement acquérir la connaissance d'adaptation auprès des experts demande un travail considérable d'extraction et de modélisation des connaissances qui peut se révéler très complexe (e.g. transfert d'expertise réalisé entre un ingénieur de la gestion des connaissances et un expert du domaine) et coûteux en temps. De plus la question de la maintenance de la connaissance acquise se pose inéluctablement. L'implémentation de ces connaissances sous forme de règles est un réel frein même si dans certains cas cette approche est améliorée par l'ajout d'une mesure de confiance sur les règles acquises [DAQ 04]. En effet, les systèmes à base de règles ont démontré leurs limites avec notamment la difficulté à gérer l'ensemble des règles produites. Après quelques tentatives, ces systèmes se sont avérés particulièrement inefficaces pour la conception en génie des procédés de par la complexité et la forte non-linéarité des phénomènes mis en jeu.

Une troisième alternative qui s'avère plutôt intéressante est celle d'acquérir ces connaissances durant le processus de résolution du problème de manière interactive. Les connaissances ainsi acquises « en ligne » ont l'avantage d'être faciles à comprendre car elles

sont souvent accompagnées d'explications. A l'inverse, les connaissances d'adaptation générées automatiquement ont l'avantage d'être facilement opérationnalisables (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). Cette stratégie présente l'avantage de situer l'utilisateur au cœur de l'adaptation et donc de pouvoir exploiter ses connaissances durant cette phase.

L'approche proposée dans ce travail, reprend cette dernière stratégie afin de réaliser le couplage entre le RàPC et l'approche par contraintes. Les PSC permettront de formaliser les connaissances présentes dans la base de cas et la stratégie d'acquisition de connaissances d'adaptation cherchera à modéliser en ligne les connaissances de l'utilisateur via une boucle d'interaction. Ces connaissances pourront être acquises sous la forme d'opérateurs permettant de modéliser les modifications à apporter à *Sol(source)*. L'ensemble de ces modifications seront regroupées dans une méthode d'adaptation qui sera attachée au cas *source* modifié.

5 Conclusion

Dans ce chapitre, les principes ainsi que les différentes approches de l'étape de l'adaptation du RàPC ont été présentés. On souligne la difficulté qu'elle représente lors de sa mise en œuvre, car indépendamment du domaine d'application, elle nécessite l'utilisation de diverses techniques et l'ajout de connaissances supplémentaires.

Après l'analyse de diverses stratégies, à la fin du chapitre une stratégie est retenue portant sur la combinaison de deux types d'approches : l'approche à base de contraintes qui formalise les connaissances présentes dans la base de cas sous la forme de PSC et une approche interactive pour acquérir directement les connaissances de l'utilisateur.

Le couplage du RàPC et de la PPC

Ce chapitre a pour but, de détailler une approche basée sur le couplage entre le RàPC et la PPC pour l'aide à la décision en conception. L'objectif est de profiter des avantages de chacun tout en minimisant leurs inconvénients. Ce couplage est réalisé grâce à une modification du cycle traditionnel du RàPC dans le but d'introduire une boucle d'interaction avec l'expert lors de l'étape d'adaptation. Cette interactivité est également exploitée pour acquérir en ligne les connaissances d'adaptation nécessaires à la conception de systèmes sans contrainte pour l'expert. L'objectif principal de l'approche proposée est de réduire l'effort d'extraction, de formalisation, de modélisation et de capitalisation des connaissances.

Chapitre 5 : Le couplage du RàPC et de la PPC

1 Introduction

Comme le chapitre 4 l'explique, la gestion des connaissances d'adaptation dans les systèmes de RàPC reste un problème complexe. Les connaissances étant souvent spécifiques à un cas ou un domaine particulier, les représenter demande un important effort d'ingénierie. La PPC offre un cadre intéressant pour atteindre cet objectif. L'articulation entre PPC et RàPC au niveau de l'étape d'adaptation permet d'exploiter la complémentarité des deux approches afin de proposer un outil d'aide à la conception. A l'heure actuelle, même si les deux approches, ont été appliquées individuellement, dans le domaine des procédés, le couplage des deux n'a jamais été étudié.

Bien que l'on s'attache à exploiter les avantages des deux approches tout en essayant d'en réduire les inconvénients, le problème d'acquisition de connaissances d'adaptation reste posé. Parmi les approches proposées dans le chapitre 4, la solution qui consiste à disposer d'un processus d'acquisition de connaissances au fur et à mesure des sessions de résolution s'avère la plus intéressante car elle est peu contraignante pour l'expert. Il s'agit de créer une interaction avec l'expert à chaque fois qu'un échec d'adaptation ou une réparation de la solution est nécessaire. Tirer profit des épisodes de résolution de problème nécessite la mise en place d'un processus d'apprentissage. L'approche proposée, s'appuie sur l'exploitation des interactions entre l'expert et le système lors de la correction d'une solution mémorisée. Lors de cette collaboration, la connaissance d'adaptation est mise à jour ou acquise (lorsqu'il s'agit de nouvelle connaissance) grâce à une modification du cycle du RàPC avec la création d'une boucle d'interaction au niveau de l'étape de révision.

2 Les principes du couplage RàPC - PPC

Après avoir présenté et analysé le RàPC (chapitre 2) et la PPC (chapitre 3), 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 deux méthodes de résolution de problème, elles reposent sur des connaissances différentes : implicites (mais explicites également possible) et contextualisées pour le RàPC, explicites et formalisées pour la PPC.

De façon générale, le développement de méthodes hybrides couplant plusieurs méthodes ou approches, est souvent envisagé afin de réduire les déficiences de l'une à l'aide

des avantages de l'autre, ou encore d'améliorer l'efficacité d'une approche en lui adjoignant de nouvelles fonctionnalités. Toutefois, même si le couplage des deux approches semble intéressant, il convient d'étudier en détail cette faisabilité d'une part et de déterminer la façon de les articuler d'autre part. Sur ce dernier point, divers travaux de la littérature présentent différentes stratégies de couplage. En effet, le couplage proposé peut également éviter des inconvénients majeurs des approches précédentes, comme par exemple le manque d'interactivité et de cohérence pour le RàPC, et l'effort conséquent d'abstraction pour la PPC. Sans traitement particulier, ces inconvénients ne seront pas résolus et limiteront l'efficacité du couplage proposé.

2.1 Une comparaison entre les deux approches

Après avoir analysé séparément le RàPC et la PPC respectivement dans les chapitres 2 et 3, et avant de présenter l'approche décrivant leur couplage, une comparaison entre ces deux approches est dressée dans le tableau suivant :

RàPC	PPC
Méthode opérationnelle.	Méthode heuristique.
Utilisation de connaissances implicites et contextualisées.	Utilisation de connaissances explicites et bien formalisées.
Raisonnement analogique dans un domaine avec un petit ensemble de connaissances.	Raisonnement basé sur des contraintes. Le nombre de celles-ci ne doit pas être ni trop restrictif (sur-contraint) ni trop extensive (sous-contraint).
Possibilité de trouver des solutions à des problèmes complexes même, si le domaine n'est pas bien connu pour l'utilisateur.	Possibilité de trouver toutes les solutions des problèmes complexes. L'utilisateur doit bien connaître la situation problématique.
Il est capable de proposer des solutions rapidement.	Pas de garantie « théorique » sur les temps de recherche.
Grâce à sa mémoire, le RàPC permet de proposer des solutions sans avoir besoin de les dériver à partir de zéro.	L'absence de mémoire oblige l'utilisateur à une analyse fine du modèle PSC en cas d'échec.
Flexibilité dans la modélisation des connaissances. L'information contenue dans les cas n'est pas forcément conditionnée à un formalisme particulier.	La richesse qu'elle offre en termes de modélisation de problèmes car l'expression des contraintes n'est pas limitée à des relations linéaires. Une syntaxe proche de l'expression du problème réel.

La limitation à des problèmes voisins et donc l'absence de garantie d'optimalité des solutions.	Possible garantie de trouver l'optimum, en introduisant une fonction objectif.
L'efficacité d'un tel système est fortement liée à la couverture de l'espace des problèmes et des solutions.	L'efficacité réside en la modélisation correcte du problème et la performance des algorithmes utilisés pour sa résolution.
Il trouvera un cas <i>source</i> plus ou moins similaire de <i>cible</i> et donc un tel système est incapable d'établir qu'un problème n'a pas de solution.	La possibilité du contrôle du nombre et de la qualité des solutions.
Possibilité d'interaction avec l'utilisateur en choisissant une stratégie appropriée d'adaptation.	Pas d'interaction lors de la résolution d'un problème. Sauf contrôle du processus au plus près des contraintes

Tableau 5.1 Comparaison entre les approches du RàPC et de la PPC

2.2 Les raisons du couplage RàPC - PPC

Les différences détaillées dans le tableau 5.1 mettent en évidence la complémentarité des deux approches. Plusieurs avantages ont été observés par [SQA 99] et [HOK 06] pour un tel couplage. Du côté RàPC l'intégration de la PPC apporte :

- l'amélioration de la formalisation des cas au travers d'une représentation plus rigoureuse grâce au formalisme PSC,
- le développement d'un mécanisme plus systématique et plus efficace de remémoration,
- une aide pour réduire la complexité par l'intermédiaire de la propagation de contraintes,
- une aide pour contrôler des préférences de conception.

Au niveau de la PPC, l'intégration du RàPC permet selon [SQA 99], [ROL 10] et [ROL 11] de :

- proposer rapidement des modèles de PSC afin d'offrir aux concepteurs une base sans avoir besoin de les dériver à partir de zéro,
- compléter des modèles lorsque la connaissance sur un domaine est floue, ou quand un modèle de contraintes n'est pas complet,
- réutiliser une expérience antérieure pour exploiter la connaissance et augmenter l'efficacité du processus de résolution,
- réutiliser des solutions de PSC,
- ajouter d'une méthode d'apprentissage au processus de PSC.

Les deux méthodes proposées possèdent donc leur propre motivation pour développer un couplage entre le RàPC et la PPC. Le but étant que l'application résultante offre des fonctionnalités et performances au-delà de ce qu'elle aurait pu apporter si l'une ou l'autre des approches avait été utilisée seule. Cependant, il subsiste des limites à ces combinaisons qui ne permettent pas d'atteindre les résultats attendus. Pour les pallier, il faudra proposer un processus d'acquisition des connaissances d'adaptation apte à offrir de la souplesse mais également capable de réduire les inconvénients d'interactivité et de cohérence du RàPC et de faciliter la mise à jour des connaissances dans la PPC.

2.3 Les limites de l'intégration du RàPC et la PPC

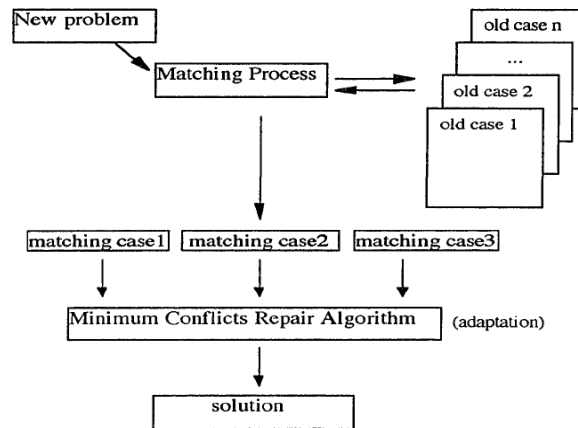
Malgré les nombreux avantages de l'intégration du RàPC et de la PPC, il subsiste certains inconvénients, comme notamment [SQA 99] :

- Le couplage des deux méthodes peut représenter un problème lorsque certains composants doivent être ajoutés (i.e. choix du solveur, de l'algorithme de résolution) lors de la mise en place dans un seul système.
- Les limites d'une méthode peuvent être héritées au système résultant du couplage surtout au niveau de l'adaptation du RàPC où les connaissances demeurent dans un domaine spécifique.
- Il peut exister des limitations par rapport au temps et à l'espace de chaque mode de raisonnement. Les méthodes de RàPC traditionnelles nécessitent plus d'espace pour stocker des cas et par conséquent plus de temps pour effectuer la remémoration. La PPC de son côté, peut ajouter plus de complexité en raison des types de problèmes traités et la performance des algorithmes de résolution utilisés.
- La mise à jour des modèles peut ne pas être appropriée dans tous les domaines, en effet, une telle mise à jour peut entraîner des inconsistances et limiter le modèle l'empêchant de résoudre des problèmes pour lesquels il a été originalement conçu.

2.4 Les travaux couplant RàPC et PSC

Plusieurs auteurs ont défendu la synergie entre RàPC et PSC, dans le but d'améliorer la qualité des solutions obtenues à partir des systèmes intégrant ces deux approches. Depuis les premiers systèmes de RàPC, comme JULIA [HIN 92], qui envisageait déjà l'incorporation de l'approche par contraintes pour la représentation des cas dans la base, et le système CADRE [HUA 96] qui intégrait l'approche par contraintes et la production de règles de connaissance

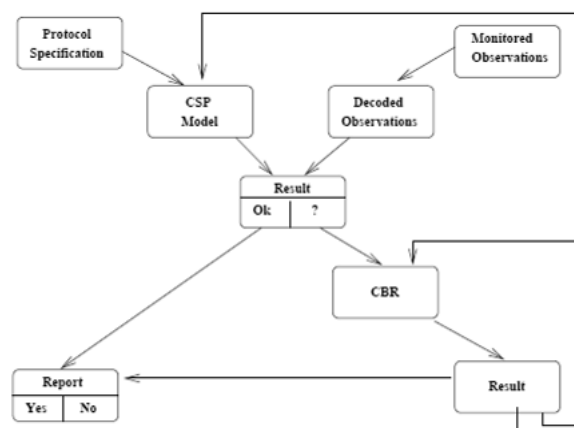
topologique. Mais également les travaux de Pu et Purvis [P&P 95], [P&P 97] et [P&P 98] dont leur système COMPOSER formalisait l'étape d'adaptation avec des contraintes en représentant les cas comme des PSC (figure 5.1). Tous ces systèmes pointaient déjà les avantages de ce couplage.



Source : [P&P 95]

Figure 5.1 Modèle d'adaptation de COMPOSER utilisant les PSC

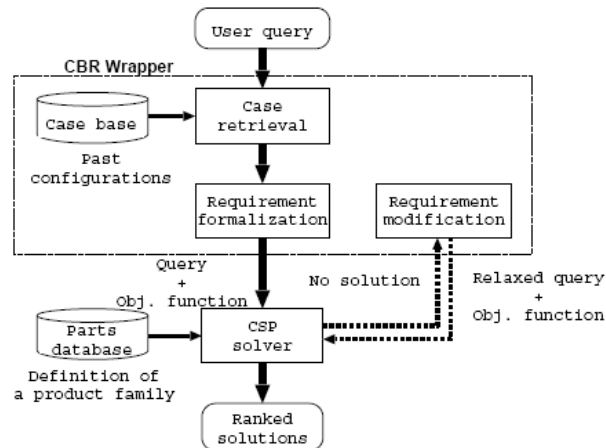
Un vaste état de l'art recensant tous ces premiers travaux sur le couplage RàPC-PPC a été réalisé par [SQA 99], [MAR 02] et plus récemment par [MAR 06]. Parmi ceux-ci on souligne les travaux de [S&F 98] où le couplage est appliqué pour compenser l'imperfection et l'inexactitude des modèles de PSC, avec une application sur les essais d'interopérabilité de protocoles de gestion de réseaux (figure 5.2).



Source : [S&F 98]

Figure 5.2 Modèle d'intégration de PSC et RàPC

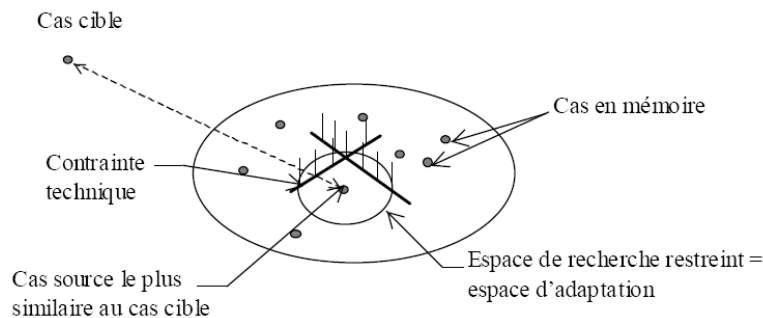
Parmi les travaux récents, ceux de Inakoshi et al. [INA 01] par exemple, proposent un cadre pour la configuration de produits. Face à un problème, ils utilisent le RàPC pour remémorer des cas similaires de configuration. Une formalisation des critères des produits est ensuite réalisée en conformité avec les préférences de l'utilisateur en s'appuyant sur les cas remémorés. Finalement le solveur PSC traite le modèle de configuration résultant comme un CSP et obtient des solutions qui sont hiérarchisées par une fonction objectif (figure 5.3).



Source : [INA 01]

Figure 5.3 Modèle RàPC-PSC pour la configuration de produits

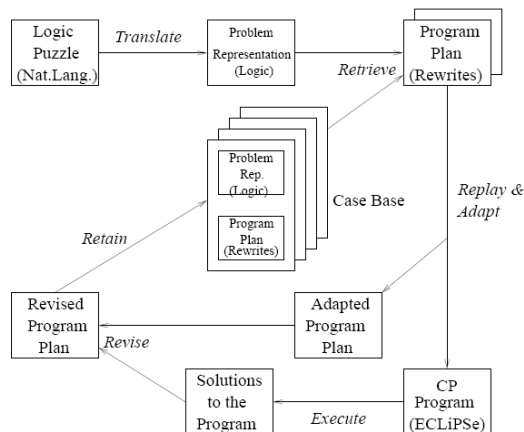
[RUE 02] combine une modélisation objet, une approche à base de cas et une approche à base de contraintes pour aider à la conception d'opérations d'usinage. Les techniques de satisfaction de contraintes sont utilisées pour guider la phase d'adaptation de la solution remémorée par le RàPC. Pour ce faire, l'approche détermine l'espace de recherche dans lequel seront appliquées les contraintes (figure 5.4). Cet espace est défini par le calcul de domaines d'adaptation pour chaque attribut caractérisant le cas *source* remémoré. Ensuite, grâce à des techniques de satisfaction de contraintes, des valeurs incohérentes et qui ne peuvent pas faire partie de la solution sont éliminées de ces domaines d'adaptation.



Source : [RUE 02]

Figure 5.4 Représentation du processus d'adaptation

Dans [LIT 02] un outil de RàPC est proposé pour stocker, remémorer et réutiliser des programmes de PPC. Le but est de rendre la PPC plus facilement accessible aux utilisateurs en accélérant la diffusion des connaissances accumulées dans un domaine (figure 5.5).

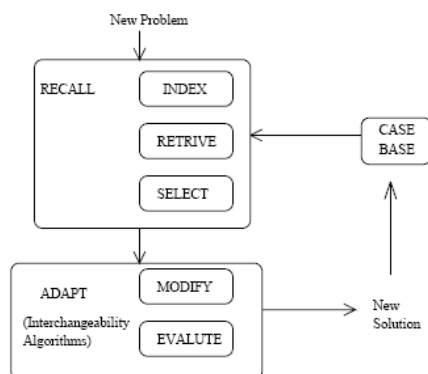


Source : [LIT 02]

Figure 5.5 Modèle RàPC pour la construction de problèmes de PPC

Lopez [LOP 03] dans le domaine d'ordonnancement, utilise le formalisme des PSC pour représenter les cas présents dans la base et le RàPC pour produire une solution, qui doit être approuvée par l'utilisateur. S'il ne la retient pas une adaptation est conduite grâce à des techniques de résolution de PSC. Dans son approche, seuls les cas qui ont été adaptés et qui ont conduit à une solution satisfaisante sont stockés dans la base, ce qui réduit l'effort de maintenance de celle-ci et lui maintient une taille raisonnable.

De son côté Neagu [NEA 05] présente de nouveaux algorithmes pour une interchangeabilité des parties des PSC afin de faciliter l'adaptation entre problèmes dans le RàPC (figure 5.6). Il propose ainsi une plateforme générique pour l'adaptation de cas où les connaissances du domaine peuvent être représentées comme des PSC. Il développe son approche pour aborder des problèmes de planning, d'ordonnancement et de configuration.

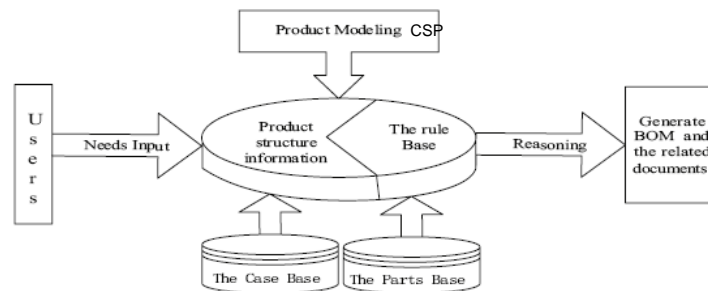


Source : [NEA 05]

Figure 5.6 Plateforme générique pour l'adaptation de cas dans le RàPC

De même, Medjdoub [MED 09] intègre les deux approches pour aborder le processus d'adaptation de problèmes dans le domaine de la conception architecturale. Son système substitue les parties d'une solution passée qui ne correspondent pas aux critères de la nouvelle requête. Le système est capable d'identifier les parties inconsistantes et grâce à un algorithme PSC, il résout ces inconsistances une à une. Cela diminue la complexité du problème.

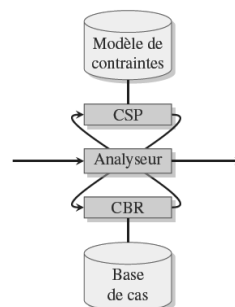
Wang et al. [WAN 09] proposent un algorithme intégrant les deux approches pour la configuration des produits « en ligne » en trois phases. L'utilisateur saisit sa requête sur la structure d'un produit puis le système RàPC cherche dans la base et récupère un cas similaire. Finalement le cas est modifié selon les contraintes, par un algorithme de configuration utilisant les PSC, pour obtenir une solution conforme à sa requête (figure 5.7).



Source : [WAN 09]

Figure 5.7 Modèle du processus de configuration de produits

Finalement [COD 10] et [VAR 11] dans le domaine de la planification de tâches (maintenance d'hélicoptères), combinent les deux approches d'une manière simultanée et itérative selon la disponibilité des connaissances prenant en compte des contraintes contextuelles issues du RàPC dans le PSC. Cela est rendu possible grâce à l'utilisation simultanée des deux approches afin que les entrées de l'un soient en partie alimentées par les sorties de l'autre (figure 5.8). Une interaction est aussi proposée afin de réduire l'espace de solutions par les entrées de l'utilisateur.



Source : [COD 10]

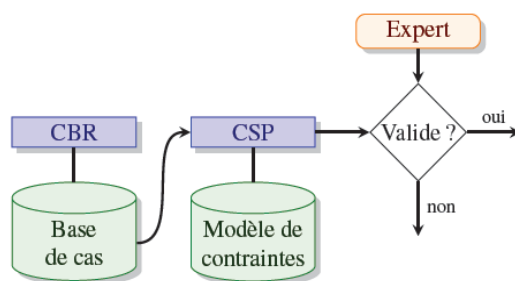
Figure 5.8 Modèle de filtrage à base de cas

2.5 L'identification des différents modes de couplage

A partir de leurs observations sur les différentes approches, Vareilles et al. [VAR 11] ont résumé les principaux modes de couplage. Ces différentes façons d'intégrer répondent à la nécessité de compléter ou de valider les connaissances capitalisées. Ces différents modes sont décrits par la suite :

1. L'emploi d'une méthode pour valider l'autre :

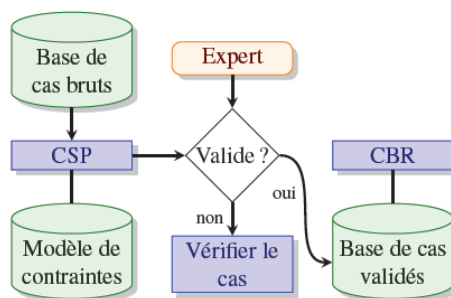
- a. *La base de cas permet de valider le modèle de contraintes (figure 5.9).* Le principe est de vérifier le modèle de contraintes grâce aux cas mémorisés. Chacun des cas est passé dans le PSC et celui-ci renvoie les inconsistances. Après avoir passé un nombre suffisant de cas dans le PSC, il est alors possible de valider ou non certaines contraintes de ce modèle.



Source : [COD 10]

Figure 5.9 Validation du PSC par le RàPC.

- b. *Le modèle de contraintes permet de valider la base de cas (figure 5.10).* Lorsque la confiance dans la connaissance stockée est accordée au modèle de contraintes, chaque cas de la base de cas peut être confronté au modèle PSC qui le validera. S'il est validé, il garde sa place dans la base de cas par contre s'il ne l'est pas, il en sera retiré afin de ne pas fausser les requêtes futures.

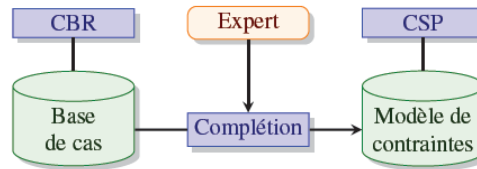


Source : [COD 10]

Figure 5.10 Validation du RàPC par le PSC

2. L'utilisation d'une approche pour compléter l'autre :

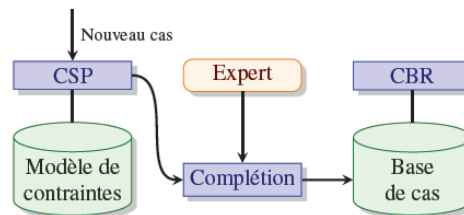
- a. *La base de cas permet de compléter le modèle de contraintes* (figure 5.11). Cela consiste à utiliser les cas présents dans la base de cas afin de compléter le modèle de contraintes du PSC. Eventuellement la base de cas peut être analysée par un expert qui peut modifier les modèles de manière « hors ligne ».



Source : [COD 10]

Figure 5.11 Analyse des cas de la base de cas afin de compléter/modifier le modèle

- b. *Le modèle de contraintes permet de compléter la base de cas* (figure 5.12). Un PSC permet de compléter les cas présents dans la base du RàPC. Cela est utile surtout lorsque la base de cas comporte des cas incomplets. L'expert peut aussi décider sur un ensemble de valeurs renvoyées par le PSC de retenir celles qui comblent cette déficience.

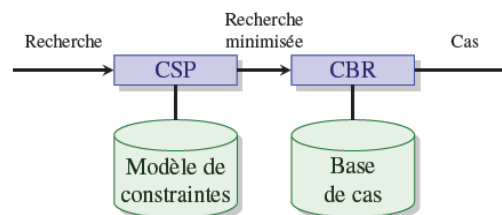


Source : [COD 10]

Figure 5.12 Utilisation du PSC pour compléter les manquements du RàPC

3. L'utilisation séquentielle des deux approches :

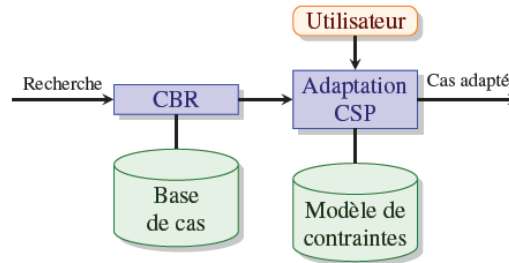
- a. *Le modèle de contraintes permet de réduire l'espace de recherche du RàPC* (figure 5.13). Cette approche vise à améliorer la performance du RàPC. A travers un premier filtrage, l'ensemble des modalités des descripteurs est filtré par le PSC, diminuant ainsi les saisies manuelles dans le RàPC. L'approche est utile lorsque certains attributs d'un RàPC peuvent être déduits à partir d'autres.



Source : [COD 10]

Figure 5.13 Réduction du domaine de recherche du RàPC par le PSC

- b. *Le modèle de contraintes aide à adapter les cas du RàPC (figure 5.14).* La principale utilisation de cette approche est axée surtout sur l'amélioration de la phase d'adaptation du RàPC. Le formalisme des PSC permet de la rendre plus abordable.

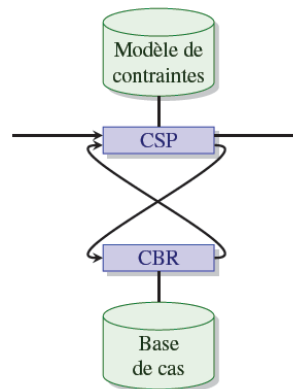


Source : [COD 10]

Figure 5.14 Utilisation du PSC pour la phase d'adaptation du RàPC

4. Le filtrage à base de cas :

- a. *Le filtrage à base de cas (figure 5.15).* Il consiste à coupler le RàPC et le PSC afin qu'ils travaillent ensemble, les entrées de l'un étant en partie alimentées par les sorties de l'autre. L'idée générale est que le PSC utilise le RàPC lorsque certaines de ses variables ne sont pas ou peu contraintes.



Source : [COD 10]

Figure 5.15 Modèle de filtrage à base de cas

- b. *Le filtrage à base de cas amélioré (figure 5.8).* Voir section 2.3

2.6 L'approche retenue

Parmi les types de couplage passés en revue dans le paragraphe précédent, *le modèle de contraintes aidant à adapter les cas du RàPC* (point 3.b) offre un cadre répondant à notre objectif : proposer une méthodologie apte à améliorer l'étape d'adaptation. En outre il offre des perspectives pour interagir avec l'expert et acquérir ainsi des connaissances sur le domaine d'application. L'approche des PSC aide à formaliser et traiter le processus

d'adaptation permettant de systématiser et de donner de la flexibilité au RàPC [PUR 98]. L'approche des PSC permet de formaliser ce processus, souvent mal défini. De plus, elle facilite la modification de la base de connaissances. Il est toujours possible d'ajouter ou d'enlever une contrainte.

Parmi les motivations pour combiner ces deux approches afin de mieux adapter les solutions de problèmes dans le RàPC, on souligne que :

- les PSC permettent une excellente représentation d'un problème sous un formalisme qui facilite son traitement durant le processus du RàPC, et plus particulièrement dans l'étape d'adaptation ;
- les PSC utilisent de puissants algorithmes pour la résolution de problèmes très complexes ;
- le RàPC est un important moyen d'apprentissage pour les PSC ;
- le RàPC augmente l'efficacité des PSC en réutilisant l'expérience passée dans le cas où la modélisation est difficile ou impossible à obtenir.

La suite de ce mémoire se concentre sur la présentation de l'approche du couplage entre le RàPC et les PSC ainsi que sur l'acquisition de connaissances « en ligne ».

3 Le couplage RàPC et PPC en conception

Dans l'approche proposée, les cas de la base sont stockés sous forme de modèles de PSC. Le principe du couplage peut se résumer de la façon suivante : le problème *cible* peut être traité par le système de RàPC qui grâce à une mesure de similarité remémore le modèle PSC d'un problème similaire (*source*). Une résolution peut être alors conduite au travers d'une technique de PPC pour réduire le domaine de validité des variables. L'ensemble des solutions possibles peut alors être déterminé.

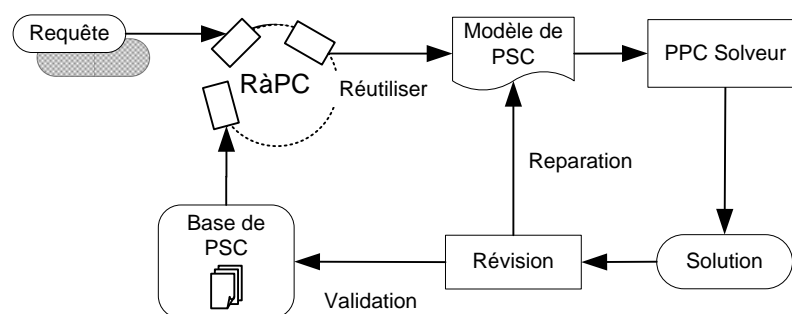


Figure 5.16 Les éléments du couplage RàPC - PPC

La figure 5.16 montre les principaux éléments de ce couplage. L'utilisation de contraintes permet de tenir compte des exigences spécifiques sur le problème rencontré. Ainsi, il sera possible de distinguer et de concaténer les exigences du domaine (contraintes des modèles) et les exigences de l'utilisateur sur un problème précis. Une révision peut être alors conduite afin que l'expert puisse éventuellement valider ces solutions. Durant cette étape l'ajout de connaissances supplémentaires est envisagé afin de corriger les échecs éventuels. Cependant on doit faire face aux problèmes d'acquisition et de formalisation. De même, les PSC nécessitent un nouveau processus pour profiter des épisodes de résolution.

3.1 La représentation des cas

L'une des premières étapes lors de la création d'un système de RàPC est celle d'établir la façon dont les cas doivent être représentés mais aussi l'information contenue dans ceux-ci. Cela permettra un meilleur traitement du problème à travers chaque étape du processus de résolution et jusqu'à l'obtention d'une solution pertinente.

Un cas est d'abord constitué par deux parties : la description du problème et un modèle PSC en guise de solution (figure 5.17). La description décrit sous la forme de paires attributs-valeurs le problème. Elle correspond à une représentation de type structurée [BER 03]. La partie solution contient un modèle de PSC qui permet de résoudre le problème ainsi décrit. Le modèle se compose d'un ensemble de variables, leurs domaines de validité et des contraintes du problème : relations logiques, expressions mathématiques...

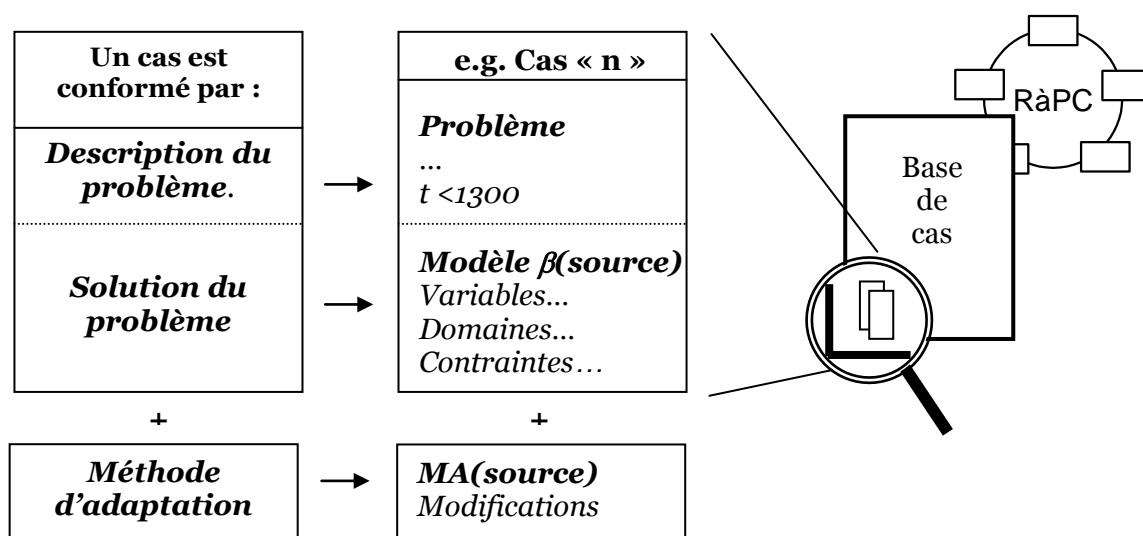


Figure 5.17 Eléments descriptifs d'un cas et de la base de cas

Dans l'approche proposée, il a été choisi de capitaliser les connaissances sur la méthode de résolution d'un problème (modèle PSC) plutôt que directement sur sa solution (capitalisation sur les β du carré d'analogie de la figure 4.2 du chapitre 4). En effet, cette méthode est souvent plus riche en connaissance, plus générique et plus facilement transposable. De plus avec l'évolution des technologies, il est plus avantageux de capitaliser sur la méthode de conception que sur le système lui-même, car la technologie peut devenir rapidement obsolète. Certes, l'évolution technologique en génie des procédés est plus lente que dans d'autres domaines mais les améliorations liées par exemple à l'intensification des procédés, génèrent de nombreux changements.

Un troisième élément peut être éventuellement présent dans la composition d'un cas : la méthode d'adaptation appelée *MA*. En effet, les modifications passées d'un cas peuvent être stockées sous la forme de méthodes d'adaptation qui sont liées au modèle β_{source} . Comme il peut y avoir plusieurs façons d'adapter un cas *source*, un cas peut posséder plusieurs méthodes d'adaptation. Ces méthodes notées $MA(source)$ étant les modifications apportées à β_{source} lors d'une résolution antérieure. Elles peuvent représenter par exemple les modifications faites par un expert lorsque son intervention est requise. Ces modifications sont stockées de façon à pouvoir être capitalisées et réutilisées dans un épisode futur.

3.2 L'acquisition de connaissances d'adaptation

L'approche de l'acquisition opportuniste des connaissances (point 3.3.4 du chapitre 4) montre l'importance d'acquérir ces connaissances durant le processus de résolution du problème : « en ligne ». Il s'agit de tirer profit des interactions de l'expert avec le système, notamment lorsqu'un échec d'adaptation se produit et qu'il est nécessaire de modifier la solution. Une fois que la solution est corrigée, les connaissances sont mises à jour ou de nouvelles sont acquises. L'avantage de cette approche est un effort réduit d'ingénierie de la connaissance par une sollicitation ponctuelle de l'expert ou l'utilisateur. De plus ce choix permet d'ajouter de l'interactivité dans le processus du RàPC et ainsi l'expert peut rapidement visualiser les conséquences de ses choix sur la solution.

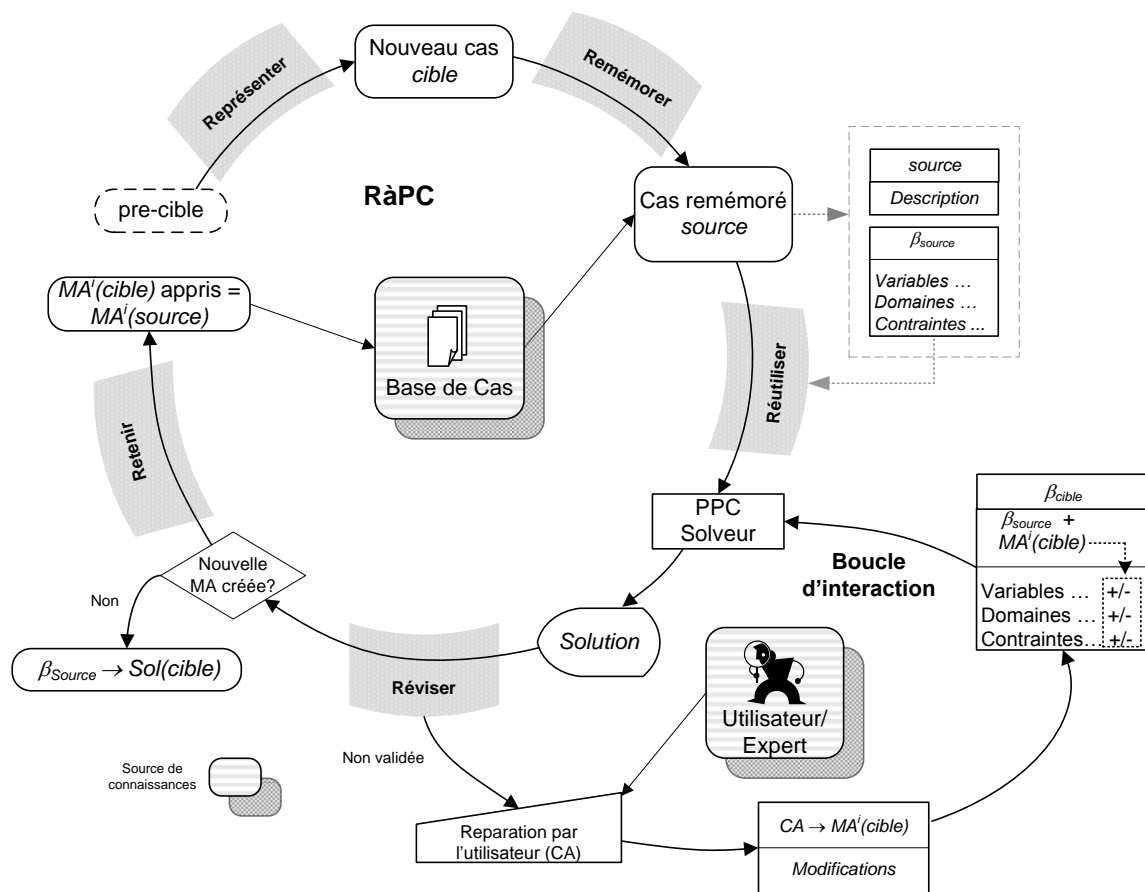


Figure 5.18 Eléments de la boucle d'interaction avec l'expert

Afin d'éviter toute ambiguïté, Il faut préciser que tout élément sortant directement de la base de cas et qui n'a pas été l'objet de modifications, est identifié avec l'appellation (*source*), de même avec tout élément qui vient s'intégrer à la base de cas. Par contre, tout élément qui a fait l'objet de modifications est identifié par l'appellation (*cible*). Par exemple, une MA remémorée de la base de cas est appelée MA(*source*). En revanche une MA qui a été créée ou modifiée, est donc identifiée comme MA(*cible*). Une fois validée et elle s'intègre la base de cas, et elle devient une MA(*source*).

La figure 5.18 introduit la nouvelle boucle dans le RàPC, durant laquelle les interactions entre l'expert et le système sont prises en compte, notamment lors de l'étape d'adaptation. Une fois le cas *source* remémoré sur la base d'une mesure de similarité, l'expert se voit présenter :

- (*source*, β_{source}) sans méthode d'adaptation créée auparavant ou s'il n'y en a aucune répondant à ses exigences, cas de figure détaillé par la suite.
- (*source*, β_{source}) avec une ou plusieurs méthodes d'adaptation MAⁱ(*source*). Lorsqu'une ou plusieurs méthodes sont envisageables, la question du choix sera abordée dans la partie

3.4 et ce cas de figure sera repris dans la partie 3.5 pour détailler la méthodologie de l'approche.

Dans le premier cas de figure, β_{source} et des connaissances supplémentaires (CA rassemblées sous la forme d'une méthode d'adaptation) sont utilisées pour construire $\beta_{cible} = Sol(cible)$, tout en s'appuyant sur la création d'une méthode d'adaptation $MA^i(cible)$ et puis retenue comme $MA^i(source)$:

Etape 1. $(source, cible) \rightarrow \Delta pb$

Etape 2. $(\Delta pb) \rightarrow \beta_{source}$

Etape 3. $(\Delta pb, CA) \rightarrow MA^i(cible)$

Etape 4. $(\beta_{source}, MA^i(cible)) \rightarrow \beta_{cible}$

Etape 5. $(cible, \beta_{cible}) \rightarrow \text{Résolution de } \beta_{cible} = Sol(cible)$

Sur l'étape 2, deux possibilités sont envisageables pour l'expert :

- β_{source} correspond à ses exigences et donc $\beta_{cible} = \beta_{source}$, il n'est pas nécessaire de créer une $MA^i(cible)$ et donc passage à l'étape 5.
- β_{source} ne correspond pas à ses exigences et il est nécessaire d'activer la boucle d'interaction pour créer une $MA^i(cible)$ et passage à l'étape 3. Cela fait appel à des connaissances supplémentaires qui seront capitalisées dans la nouvelle méthode ainsi créée. A noter que ce cas de figure n'est possible que lorsque la boucle d'interaction avec l'expert est activée. En effet lors du premier passage dans le cycle, l'expert ne sait pas a priori si β_{source} corrigée par $MA^i(cible)$ répond à ses exigences tant que l'étape 4 n'a pas été réalisée.

Après la création d'une nouvelle $MA^i(cible)$ sur l'étape 3, deux possibilités sont envisageables pour l'expert :

- β_{source} corrigée avec $MA^i(cible)$ répond à ses attentes, par conséquent passage à l'étape 5 avec $MA^i(cible)$ qui devient $MA^i(source)$ de par son intégration à la base de cas.
- β_{source} corrigée avec $MA^i(cible)$ ne donne pas satisfaction à l'expert, dans ce cas, la boucle est relancée et $MA^i(cible)$ modifiée.

3.3 L'interaction avec l'expert

Le déclenchement de la boucle d'interaction lors de la réparation d'un cas, est l'occasion d'acquérir les connaissances d'adaptation. L'ensemble de ces connaissances est

regroupé et capitalisé sous la forme d'une méthode d'adaptation. L'idée est d'exploiter ces corrections apportées dans le processus interactif.

Afin de mettre en œuvre l'approche proposée, il convient de poser l'hypothèse : les modifications peuvent être apportées par étapes élémentaires successives. Comme le propose [COR 07], chaque étape correspond à une opération d'adaptation élémentaire qui est réalisée par un opérateur d'adaptation OA . Une méthode d'adaptation est donc composée d'un ensemble fini de m opérateurs d'adaptation successifs :

$$MA^i = \{OA_i^j\} \quad i \in \{1, \dots, m^i\}$$

m^i étant variable d'une méthode à l'autre, car l'adaptation peut nécessiter plus ou moins de modifications. Ces opérateurs symbolisent les actions que l'expert réalise sur β_{source} afin d'obtenir une solution satisfaisante.

Comme chaque modèle est formulé selon l'approche de PPC qui prend en compte les variables, les domaines et les contraintes, les actions de l'utilisateur se limitent à modifier ces trois éléments sur le modèle. Les opérateurs dans une méthode d'adaptation traduisent donc les actions possibles suivantes :

- Ajouter : une nouvelle variable, une nouvelle contrainte, une valeur dans un domaine.
- Modifier : le domaine d'une variable, une contrainte, une valeur dans un domaine.
- Effacer : une variable, une contrainte, une valeur dans un domaine.

Après chaque opérateur d'adaptation créé, l'utilisateur a la possibilité de l'enrichir en ajoutant des commentaires. Ces commentaires peuvent expliquer l'intérêt de l'opérateur d'adaptation (une connaissance commentée est souvent plus crédible et réutilisable). De la même manière qu'une méthode d'adaptation peut contenir plusieurs de ces opérateurs, plusieurs méthodes d'adaptation peuvent être associées à un modèle complétant ainsi la base de cas dans le système RàPC (voir figure 5.19).

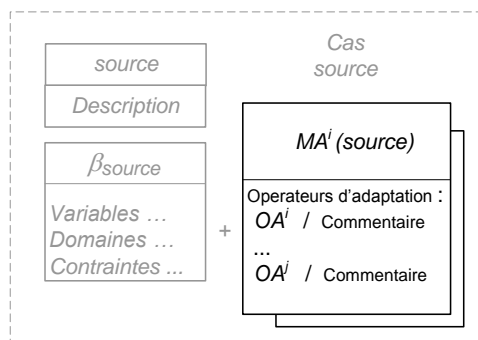


Figure 5.19 La représentation des connaissances d'adaptation dans un cas

3.3.1 La validation des opérateurs d'adaptation

Suite à l'étape de remémoration, une procédure de test calquée sur celle de [COR 07] peut être mise en œuvre afin de découvrir et, si l'utilisateur le veut, corriger le ou les opérateurs d'adaptation « défectueux » par rapport aux besoins du problème. La procédure teste successivement et séparément les différents opérateurs d'adaptation. A chaque fois qu'un opérateur est sélectionné, un nouveau modèle pour *cible* est résolu et sa solution présentée à l'expert. S'en suivent deux cas de figure :

- l'expert valide l'opérateur testé : il peut passer alors à un autre opérateur ;
- l'expert ne valide pas l'opérateur testé : le système lui demande de le corriger jusqu'à l'obtention d'un opérateur valide. Une fois corrigé et validé le système lui demande de commenter et d'expliquer cette modification.

Le système sort de cette procédure si après le test-correction d'un ou de tous les opérateurs d'adaptation, un modèle β_{cible} satisfaisant est trouvé. Dans le cas où après avoir testé tous les opérateurs, et modifié certains d'entre eux, le modèle n'est toujours pas satisfaisant, la procédure de test reste toujours active pour que l'expert puisse ajouter des opérateurs supplémentaires si nécessaire. La figure 5.20 résume l'approche.

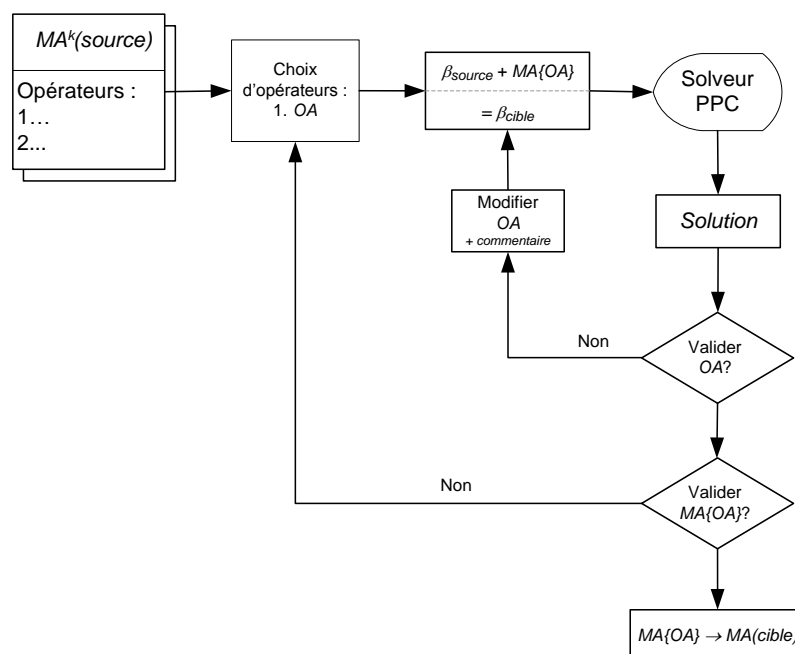


Figure 5.20 Procédure de test pour les opérateurs d'adaptation

1. Choix d'un opérateur d'adaptation
2. Construction de β_{cible} à partir de β_{source} et $MA = \{OA_j\}$ j : ensemble des opérateurs déjà testés et conservés
3. Résolution de β_{cible}
4. La solution est soumise à l'expert pour validation :
 - a. L'expert valide la solution intermédiaire mais en plus il la considère comme répondant à ses exigences initiales, alors sortie de la boucle (passage à l'étape « Retenir » du RàPC)
 - b. L'expert valide la solution intermédiaire mais elle ne répond pas complètement au problème initial (les conséquences du choix de cet opérateur restent cohérentes avec les exigences globales) alors retour au pas 1 avec mise à jour de $MA\{OA\}$
 - c. L'expert ne valide pas la solution :
 - Modification(s) de l'opérateur testé jusqu'à l'obtention d'une solution valide avec ajout d'un commentaire pour expliquer la correction (retour à l'étape 2 à chaque modification)
 - Si tous les OA ont été testés, il faut ajouter des opérateurs à la méthode d'adaptation en cours jusqu'à atteindre une solution adéquate et retour à l'étape 2.

3.4 L'évaluation des méthodes d'adaptation

Lorsqu'une méthode d'adaptation a mené à une solution satisfaisante, une évaluation peut être mise en place afin d'établir un système de recommandation pour les méthodes d'adaptation. Cette évaluation s'inspire des idées sur l'intelligence collective et le partage de connaissances abordées dans la section 3.3.7 du premier chapitre. En effet, lorsqu'un utilisateur a obtenu une solution grâce à une méthode d'adaptation, il peut donner son avis et qualifier la performance de la méthode grâce à certains critères. Les critères utilisés pour évaluer les méthodes d'adaptation sont inspirés de l'approche de Vernat [VER 04] sur l'évaluation de modèles de conception :

- sa parcimonie : qui représente la capacité à obtenir une solution avec un nombre minimum de changements sur les paramètres du modèle ;
- sa précision : qui définit l'étendue de solutions possibles obtenues pour un modèle.

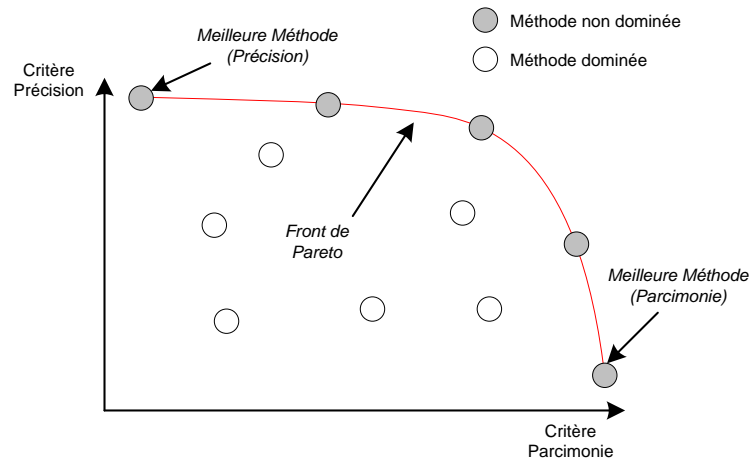


Figure 5.21 Analyse Pareto pour la qualification des méthodes

Puisque ces deux critères peuvent être difficiles à hiérarchiser, la technique du front de Pareto est envisagée afin d'aider au choix d'une méthode d'adaptation. L'utilisation de la notion de dominance de Pareto permet de présenter à l'utilisateur l'ensemble de méthodes offrant les meilleurs compromis entre les critères de précision et de parcimonie. Les deux critères sont représentés graphiquement sur un axe séparé et les différentes méthodes d'adaptation peuvent être visualisées dans un nuage de points (figure 5.21). Seuls les points non dominés peuvent être suggérés à l'utilisateur. D'après le concept du front Pareto, une méthode a domine une autre méthode b si :

- La méthode a n'est pas moins bonne que la solution b pour tous les critères ;
- La méthode a est strictement meilleure que la méthode b pour au moins un des critères.

Eventuellement des commentaires peuvent y être inclus qui unis aux évaluations proposent alors un système de recommandations utiles pour les futures consultations.

3.5 La méthodologie du couplage RàPC-PPC

Les sections précédentes montrent d'une manière générale les différentes parties comprenant l'approche proposée d'adaptation. La figure 5.21 illustre le processus d'adaptation intégré au RàPC mais cette fois-ci avec des méthodes d'adaptation remémorées avec le cas *source*.

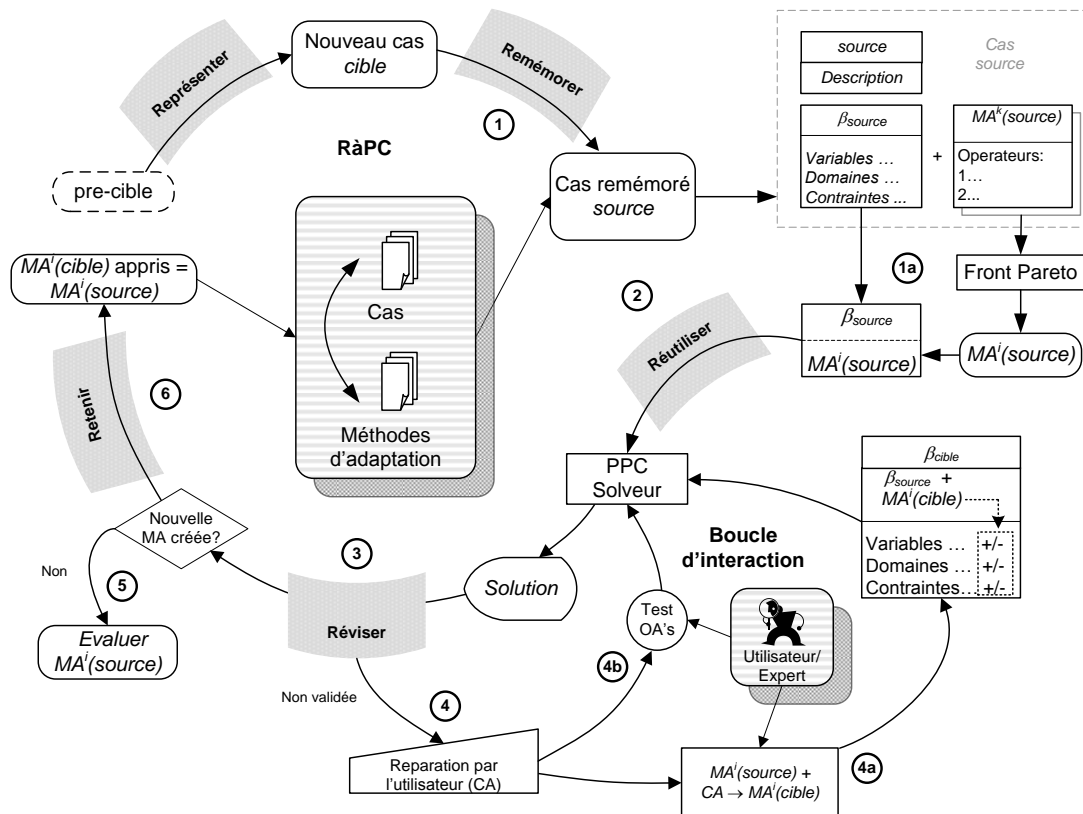


Figure 5.22 Le processus complet d'adaptation

Les différentes étapes sont décrites par la suite :

1. Remémoration du cas « source »

Après la représentation du problème *cible* et la remémoration d'un cas similaire *source* de la base des cas, le système RàPC récupère un cas « source » comprenant un modèle β_{source} avec une ou plusieurs méthodes d'adaptation associées.

a. Lorsque le cas *source* présente plusieurs méthodes d'adaptation $MA^k(source)$, le système peut lui proposer une méthode $MA^i(source)$ en se basant sur une analyse front Pareto (point 3.4).

2. Réutilisation de $\beta_{source} + MA^i(source)$

Le modèle β_{source} modifié par $MA^i(source)$ est envoyé au solveur de PPC qui résout le modèle modifié et fournit une solution.

3. Révision de la solution

Si la solution satisfait l'utilisateur il faut aller à l'étape 6 ; sauf si β_{source} a été modifiée par une $MA^i(source)$ qui n'a subi aucune modification (directement extraite de la base de cas) alors il faudra aller à l'étape 5 et mettre à jour la base de cas.

Si en revanche, la solution n'est pas satisfaisante la boucle d'adaptation est activée et passage à l'étape suivante.

4. Activation de la boucle d'interaction

Un processus est déclenché pour corriger la solution non satisfaisante avec deux possibilités :

- a. L'utilisateur modifie le modèle en créant des *OA*. L'ensemble des *OA* devient une nouvelle méthode d'adaptation $MA^i(cible)$. A noter que ces modifications sont complémentaires au $MA^i(source)$. En effet, la nouvelle $MA^i(cible)$ est générée sur la base de $MA^i(source)$ utilisée au pas 2 ou bien d'une méthode provenant d'un cycle d'adaptation antérieure. Les modifications sont donc appliquées à β_{source} qui devient un modèle β_{cible} . Ce dernier est résolu par le solveur PPC et l'utilisateur revient à l'étape 3, pour la validation du nouveau modèle.
- b. L'utilisateur opte pour vérifier les *OA* de la méthode $MA^i(source)$. La procédure de test des *OA* est déclenchée (figure 5.20). La fin de la procédure le conduit à l'étape 6.

5. Evaluation de la méthode d'adaptation.

Lorsqu'une solution satisfaisante a été trouvée grâce à $MA^i(source)$ alors une évaluation peut être faite grâce aux critères expliqués dans la section 3.4 avec également la possibilité de commenter la méthode d'adaptation.

6. Mise à jour de la base de cas

Une association entre la méthode d'adaptation $MA^i(cible)$ et le modèle β_{source} est réalisée. La base de cas est mise à jour.

Cette méthodologie sera implémentée dans le système RàP⁴C et testé avec un problème de conception de mélangeurs industriels.

4 Conclusion

Ce chapitre présente l'approche proposée pour améliorer une des étapes cruciales dans l'implémentation du RàPC : l'adaptation. Le RàPC a fait ses épreuves en tant que méthodologie très performante pour la résolution de problèmes dans divers domaines d'application et il est rapidement devenu l'une des techniques les plus intéressantes pour assister la conception. La méthodologie propose d'abord une intégration du RàPC avec l'approche par contraintes, puisqu'elle facilite le traitement des problèmes en conception. En formulant les cas comme des modèles de PSC on peut systématiser son traitement tout au long des étapes de RàPC. Cette intégration du RàPC et PSC est particulièrement performante au niveau de l'adaptation de cas. Sur cette base l'acquisition de connaissances d'adaptation est conduite grâce à une interaction « en ligne » avec l'expert. Le système proposé est

organisé et implémenté de telle sorte que l'utilisateur garde le contrôle des opérations durant l'adaptation. Ces expériences sont ainsi stockées et mises à disposition grâce à une méthode d'adaptation. L'évaluation de ces méthodes permet aussi d'inclure un système de recommandation pour faciliter encore plus les consultations futures. Les principales limites de l'approche présentée se résument à :

- La constitution de la base de cas initiale est réalisée par l'expert hors du RàPC. Par conséquent la création des premiers modèles de PPC hérite des inconvénients énoncés dans le chapitre 3.
- La base de cas ne contient pas les solutions du modèle. Pour pouvoir afficher les solutions il faut lancer le solveur PPC après chaque remémoration. Si un modèle est complexe, cela pourrait prendre un temps important pour le résoudre. Cet inconvénient, est dû au choix : de capitaliser la connaissance sur la méthode de conception et non pas sur la technologie, de diminuer la taille de la base de cas pour un gain de temps lors des remémorations. En plus, vu que les modèles peuvent présenter des nombreuses solutions, leur stockage peut devenir difficile à gérer.

Pour les suites à donner à ce travail, plusieurs perspectives sont envisageables :

- Étendre l'évaluation aux modèles de PPC β_{source} de la même manière que pour les MA. Dans l'approche seules les CA sont évaluées (OA ou MA). Mais on peut transposer ces évaluations et systèmes de recommandation aux cas : critères d'évaluation, analyse par front de Pareto. On pourrait également ajouter des critères lors de la remémoration afin de personnaliser cette étape de remémoration.
- Lorsque un cas présente plusieurs MA la question de la maintenance se pose inéluctablement. Cela peut demander un processus supplémentaire d'abord pour établir les critères pour pouvoir créer un nouveau cas à partir des MA d'un modèle β_{source} (par exemple l'obsolescence de technologie, ou des opérations ne répondant plus aux besoins). Une piste peut être envisagée en analysant les OA commun à plusieurs MA et ainsi les inclure dans le modèle β_{source} pour produire un nouveau cas.
- Finalement l'approche ne doit pas se limiter aux problèmes formulés comme des PSC car les problèmes de conception ne peuvent pas être tous formulés selon cette approche. Il faut ainsi transposer l'approche à d'autres systèmes de RàPC mais à la condition d'étendre les possibilités des OA. Dans le système ReMSiProc présenté au chapitre 2, les OA pourraient par exemple s'appliquer à la modification des opérations unitaires : changement du type de séparation, remplacement d'un réacteur et d'une séparation par une opération intensifiée...

Contexte applicatif : Le système RàP⁴C

Dans ce chapitre on présente l'application de la méthodologie proposée au chapitre précédent. Elle est mise en œuvre dans le système prototype RàP⁴C. Un cas d'étude sur la conception d'un système d'agitation illustre certaine partie de la méthodologie et démontre son intérêt pour la conception préliminaire des procédés.

Chapitre 6 : Contexte applicatif : Le système RàP⁴C

1 Introduction

Le système RàP⁴C (**R**aisonnement **à** **P**artir de **P**roblèmes de **P**rogrammation **par** **C**ontraintes) est un outil d'application de RàPC pour assister la conception dans le domaine du génie des procédés. Ce système met en œuvre la méthodologie présentée au chapitre précédent.

RàP⁴C a été conçu avec une interface graphique facilitant ainsi à l'utilisateur le processus de résolution à des problèmes de conception. Sur l'aspect technique RàP⁴C a été développé avec une approche de programmation objets utilisant le langage C++. Au niveau de l'étape de résolution, le système intègre les solveurs de PPC ILOG ou Real Paver. Dans cette première version de RàP⁴C l'accent a été particulièrement mis sur l'étape d'adaptation. Dans les sections suivantes les parties principales du système RàP⁴C sont détaillées.

2 L'architecture de RàP⁴C

La figure 6.1 offre une vue globale de l'architecture de RàP⁴C. Il possède une interface graphique intégrant plusieurs modules pour chaque étape de la méthodologie expliquée en chapitre 5. Comme dans le prototype ReMSiProc, l'utilisateur peut introduire facilement un problème au travers des différents composants. L'application possède un navigateur pour se déplacer toute au long des modules et elle permet également de consulter l'aide du système.

Après la définition du problème (Requête), le module de Remémoration lance l'algorithme de similarité (figure 2.6 du chapitre 2) pour la recherche de cas sources similaires et l'affichage des résultats de la recherche. Après la sélection d'un cas le système le résout grâce au solveur PPC et affiche la solution (Réutilisation). L'utilisateur la vérifie et lance le module d'Adaptation si des modifications doivent être apportées (Révision, boucle d'interaction).

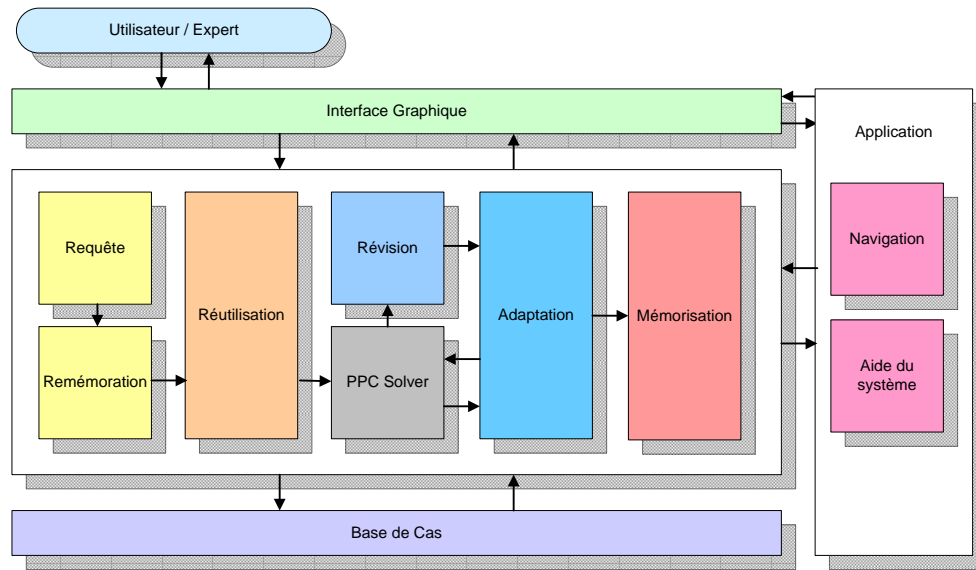


Figure 6.1 Architecture de RàP⁴C

2.1 Le développement de l'outil RàP⁴C

Pour représenter les différentes activités de RàP⁴C on emploie les diagrammes de cas d'utilisation du Langage de Modélisation Unifié (UML) de par sa simplicité de représentation des activités des différents modules d'un système et afin de faciliter le développement de l'outil et sa compréhension.

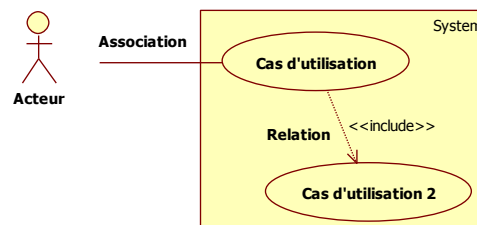


Figure 6.2 Eléments d'un diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation offrent une vision globale du comportement fonctionnel d'un système logiciel. Ils intègrent les éléments suivants : cas d'utilisation, acteurs et des relations. Un cas d'utilisation représente une description des opérations que l'acteur réalise dans le système. Les acteurs sont des entités (e.g. utilisateur, expert et système) qui interagissent avec les cas d'utilisation. Finalement les relations représentent les interactions possibles entre les éléments du diagramme. La figure 6.2 schématise les éléments des diagrammes de cas d'utilisation utilisés dans la description des étapes de RàP⁴C.

2.1.1 L'accès au système

Dans le cas d'utilisation pour l'accès au système (figure 6.3), on présente deux activités possibles : le démarrage d'une session RàPC en tant qu'utilisateur/expert ou la configuration du système en tant qu'expert. La première option permet d'accéder au système de RàPC pour saisir une requête, tandis que la seconde ouvre une session de configuration du système.

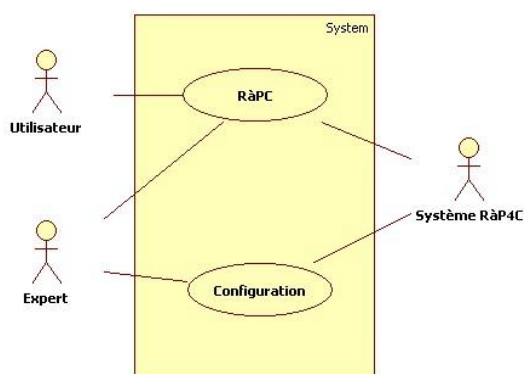


Figure 6.3 Cas d'utilisation de l'accès du système

2.1.2 La remémoration

Dans le premier module Remémoration (figure 6.4), une nouvelle requête peut être introduite. De la même manière que dans le système ReMSiProc, la seule saisie de quelques valeurs des variables permet d'extraire un cas *source* de la base de cas.

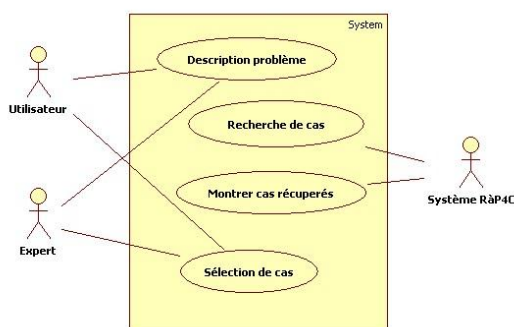


Figure 6.4 Cas d'utilisation de la remémoration

Ensuite, l'algorithme de recherche, trouve parmi les cas le plus semblable. Le système donne à l'utilisateur le choix du type de fonction de similarité à utiliser pour effectuer la remémoration. Les cas *source* remémorés sont hiérarchisés dans un tableau affichant : le degré de similarité et une visualisation du modèle β_{source} . Finalement l'utilisateur choisit un cas *source* pour sa réutilisation.

2.1.2.1 La base de cas

La figure 6.5 montre la constitution de la base de cas. Elle repose sur deux dossiers, l'un pour stocker les cas et l'autre pour stocker les méthodes d'adaptation. Dans le dossier « Cas » se trouvent les problèmes modélisés sous le formalisme PSC. Le dossier « Méthodes » contient plusieurs sous-dossiers créés lors des épisodes précédents d'adaptation où l'on retrouve les méthodes d'adaptation pour chaque cas. Chaque modèle et méthode d'adaptation sont gérés au moyen de fichier texte, supportant les formats .rtf ou .txt.

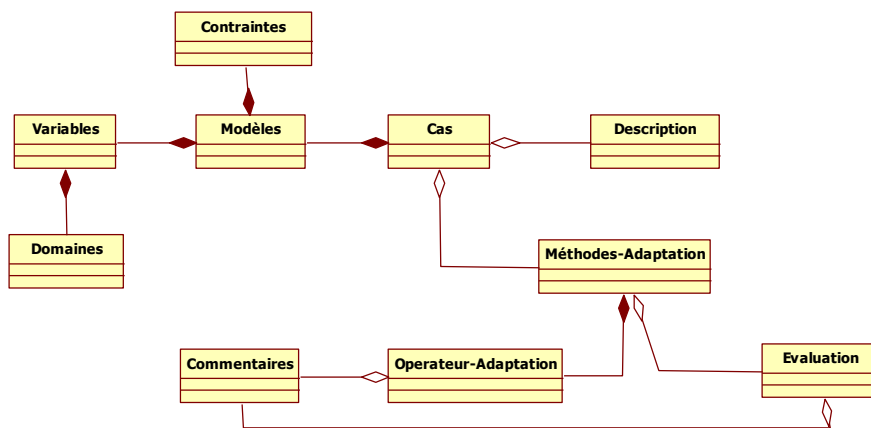


Figure 6.5 Structure de la base de cas dans RàP⁴C

2.1.3 La réutilisation

Suite à l'étape de remémoration, le module de Réutilisation récupère le cas *source* et affiche ses différents éléments : la description du problème, le modèle β_{source} et éventuellement des méthodes d'adaptation. Le module affiche les éléments de β_{source} , (ensemble de variables, domaines de définition et contraintes) sur de composants différents en respectant sa mise en forme. Par ailleurs, une identification des deux types de contraintes a été proposée :

- les contraintes génériques : les contraintes portant sur les connaissances génériques. Elles représentent les contraintes dures sur les aspects généraux sur le système à concevoir.
- les contraintes spécifiques : ces contraintes sont spécifiques à la définition du problème (exigences du point de vue du concepteur), ces contraintes souples liées aux spécifiées du problème traité.

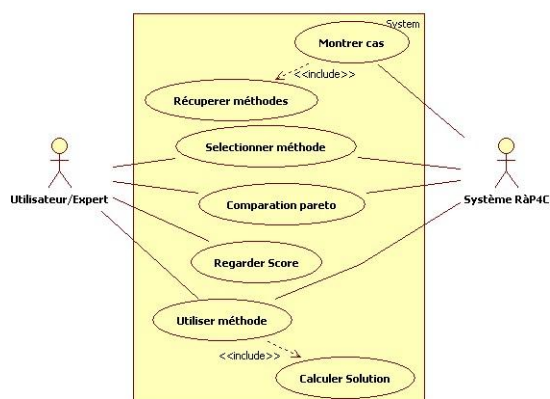


Figure 6.6 Cas d'utilisation de la Réutilisation

2.1.3.1 Les méthodes d'adaptations associées

Lorsque β_{source} possède des méthodes d'adaptation $MA^i(source)$, elles sont remémorées et présentées à l'utilisateur. Ces méthodes sont affichées dans une liste et leurs operateurs d'adaptation sont déployés dans un composant à côté. Si l'utilisateur veut utiliser une méthode, il peut la sélectionner et appliquer les changements au modèle récupéré. Si une méthode d'adaptation $MA^i(source)$ est choisie, le nouveau modèle β_{cible} est envoyé au solveur PPC pour procéder au calcul de la solution en prenant compte les modifications.

Pour aider l'utilisateur au choix d'une méthode d'adaptation, l'analyse du front de Pareto est utilisée. La figure 6.7 donne un aperçu de cette implémentation. Une comparaison de critères Pareto « critère versus critère » est affichée grâce à la fonction « Comparaison Pareto ». Pour l'analyse, toutes les méthodes d'adaptation ayant fait l'objet d'une évaluation sont considérées.

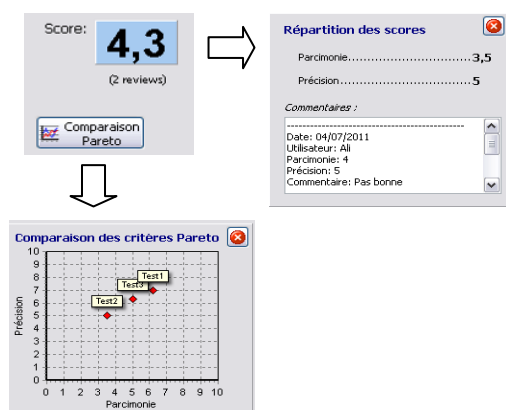


Figure 6.7 Le système de recommandation dans l'étape de réutilisation

2.1.4 La Révision

Suite à l'étape de réutilisation, le modèle β_{source} est résolu par le solveur de PPC. Dans ce module les actions possibles peuvent être : la révision des solutions, l'évaluation d'une méthode $MA^i(source)$ ou la mémorisation de la méthode d'adaptation $MA^i(cible)$.

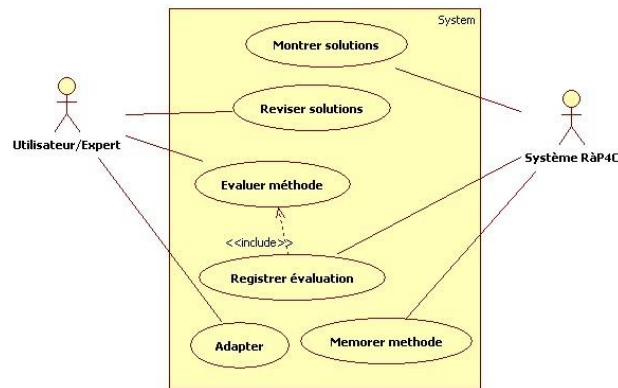


Figure 6.8 Cas d'utilisation de la solution

2.1.5 L'Adaptation

Le module d'adaptation est initié lorsque la solution trouvée à l'étape précédente ne correspond pas aux attentes de l'utilisateur. La figure 6.9 ci-dessous décrit le module d'adaptation et les différentes parties qui le composent. Ensuite la figure 6.10 illustre le cas d'utilisation pour la procédure de tests des OA.

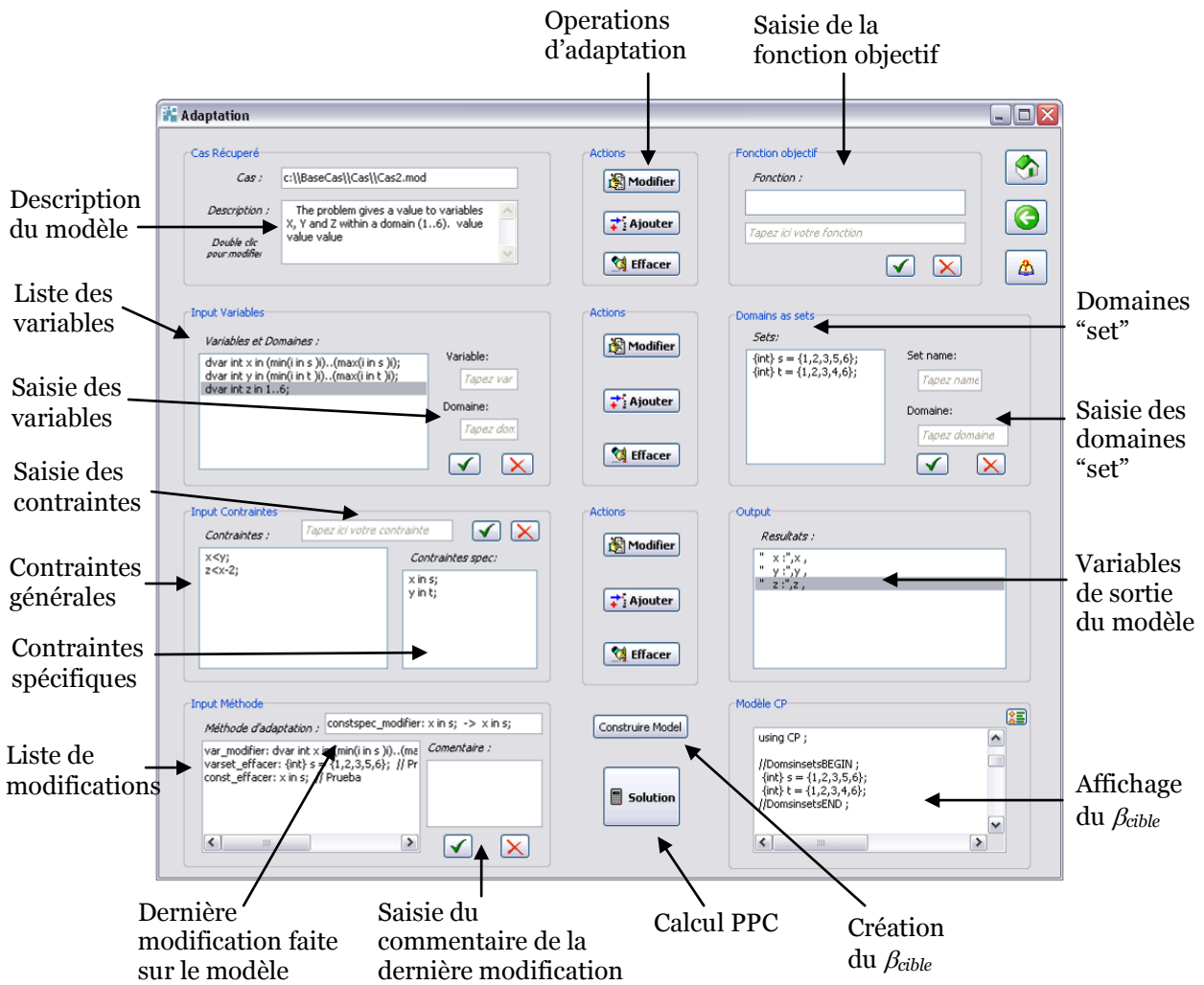


Figure 6.9 Le module d'adaptation

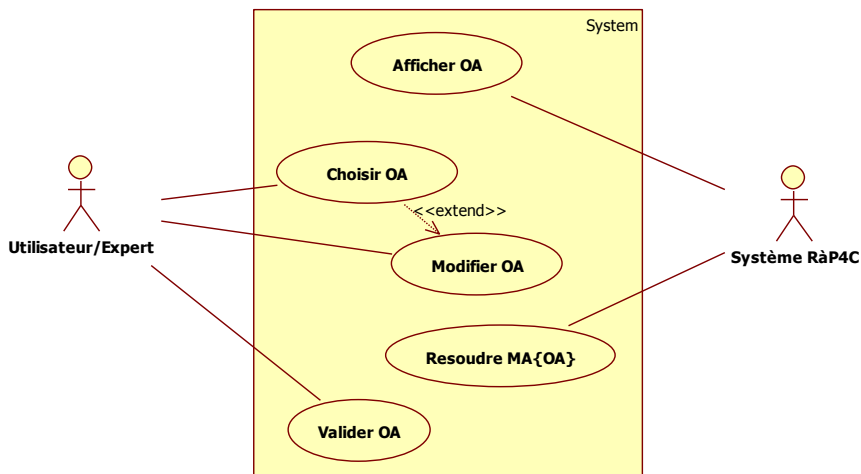


Figure 6.10 Cas d'utilisation de la procédure de test des OA

2.1.5.1 Les modifications de l'utilisateur

Lorsque le module d'adaptation est activé, le modèle β_{source} est visualisé dans le module « Adaptation ». Chaque composant listant les variables, domaines et contraintes possède des boutons associés, permettant à l'utilisateur trois actions : modifier, ajouter ou supprimer. Les boutons de modifications utilisent des messages d'alerte pour guider l'utilisateur pendant les changements à effectuer. Chaque modification est alors déclinée dans un composant pour garder une trace des changements. Les modifications sont traduites à l'aide d'opérateurs prédéfinis afin d'identifier les actions faites par l'utilisateur. L'ensemble des opérations possibles et un exemple de leur syntaxe sont donnés dans les tableaux suivants :

OA Variables	Syntaxe
Modifier variable	<code>var_modifier: dvar int z in 1..6; -> dvar int z in (min(i in t)i)..(max(i in t)i);</code>
Ajouter une variable	<code>var_ajouter: {string}TypeC={"Carre", "Cylin", "Hemis"};</code>
Effacer une variable	<code>var_effacer: dvar int x in 1..6;</code>

Tableau 6.1 OA de variables

OA Domaines	Syntaxe
Modifier un domaine	<code>domaine_modifier: t in {1,2,3,4,6}; -> t in {3,4,5,7,9};</code>
Ajouter une valeur au domaine	<code>domaine_ajouter: TypeC + {"Parallelel"};</code>
Effacer une valeur au domaine	<code>domaine_effacer: TypeC - {"Carre"};</code>

Tableau 6.2 OA de domaines

OA Contraintes	Syntaxe
Modifier une contrainte générale	<code>const_modifier: x<y; -> x>y;</code>
Ajouter une contrainte générale	<code>const_ajouter: z-x;</code>
Effacer une contrainte générale	<code>const_effacer: z-x;</code>
Modifier une contrainte spécifique	<code>constspec_modifier: x-y; -> x+y;</code>
Ajouter une contrainte spécifique	<code>constspec_ajouter: x-y;</code>
Effacer une contrainte spécifique	<code>constspec_effacer: x+y;</code>

Tableau 6.3 OA de contraintes

OA Fonction objectif	Syntaxe
Modifier une fonction objectif	fonction_modifier: maximize E; -> minimize E;
Ajouter une fonction objectif	fonction_ajouter: minimize coût;
Effacer une fonction objectif	fonction_effacer: maximize D;

Tableau 6.4 OA de fonction objectif

Lorsqu'un problème d'optimisation se présente, le module permet de modifier cette fonction de la même manière que les autres éléments du modèle (voir tableau 6.4 ci-dessus).

Après chaque modification une saisie de commentaires peut être faite dans la partie « Input Méthode ». Les commentaires sont ainsi stockés une fois que la méthode est créée. Ce commentaire est inséré à la suite de la modification de la façon suivante :

const_modifier: x<y; -> x<y; // partie commentaire débutant par un //

2.2 Sortie de la session

L'étape de mémorisation est activée lorsque l'épisode de résolution réussi à trouver une solution qui satisfasse l'utilisateur. Cette dernière étape du processus peut également être lancée depuis l'étape de réutilisation, lorsqu'une solution y est trouvée ou bien quand le processus d'adaptation a produit une solution.

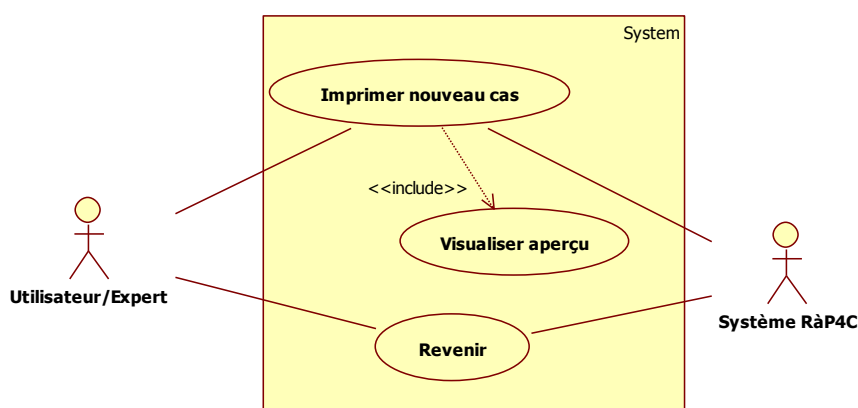


Figure 6.11 Le cas d'utilisation du module Remémoration

3 Exemple d'application : configuration d'un dispositif agitateur

L'étude des systèmes liés à la conception de procédés représente la principale application de ce travail de thèse, néanmoins pour mettre au point l'outil et afin de valider le fonctionnement du système RàP⁴C, une application est ici proposée au travers d'un exemple sur la conception d'une opération unitaire.

3.1 Introduction

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, création d'une émulsion... 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.

3.2 La nature des phases et les opérations

Une classification des opérations d'agitation les plus courantes est dressée dans le tableau 6.4. Elles sont classées en fonction du type de fluides concernés. Deux distinctions peuvent y être faites si l'on regarde le rôle que doit jouer l'agitateur lors de ces différentes opérations. Celles qui nécessitent du mouvement (e.g. homogénéisation, suspension). Cette catégorie concerne le cas des liquides miscibles et des systèmes liquide-solide. Ensuite il y a les opérations qui impliquent la génération de la dispersion d'une phase dans une autre (e.g. émulsification, dispersion). Cette deuxième catégorie se rapporte plutôt aux systèmes liquide-gaz et liquides non miscibles. Dans toutes les opérations un transfert thermique peut être réalisé par la présence éventuelle d'un dispositif supplémentaire pour un tel effet. En général, les opérations imposent des conditions particulières qui déterminent le choix d'un agitateur pour une opération spécifique. La difficulté réside dans la sélection d'une gamme d'équipement la mieux adaptée à l'objectif désiré [XUE 06].

Nature des phases	Phénomène hydromécanique	Opération	
Liquides miscibles	Homogénéisation Maintien en suspension	Réaction chimique	Transfert thermique
Liquides immiscibles	Dispersion Emulsion	Réaction hétérogène Extraction	
Liquide-solide	Mise en suspension Maintien homogène Délitage, dispersion	Dissolution Cristallisation Précipitation Élutriation	
Liquide-gaz	Dispersion	Absorption Désorption	

Source : [XUE 06]

Tableau 6.5 Quelques opérations nécessitant de l'agitation

3.3 Les caractéristiques hydrodynamiques

Les propriétés des écoulements observés dans une cuve agitée sont à la base des performances de l'équipement. Elles vont permettre une partie des choix et du dimensionnement des installations, en complément de l'analyse énergétique du système. Les caractéristiques suivantes ont été choisies pour exemple :

- **Pompage** : le pompage est la grandeur de base qui va définir les performances d'un système. Le pompage va quantifier la capacité du système à générer plus ou moins de mouvement.
- **Cisaillement** : lorsqu'un fluide est mis en mouvement, on obtient un champ de vitesse qui traduit le comportement de ce fluide. Entre deux points adjacents du volume concerné, il existe généralement un gradient de vitesse. Les forces de frottement exercées par deux couches infinitésimales de fluide glissant l'une par rapport à l'autre à des vitesses différentes sont appelées les forces de cisaillement.
- **Turbulence** : la turbulence désigne l'état d'un fluide, liquide ou gaz, dans lequel la vitesse présente des tourbillons dont la taille, la localisation et l'orientation varient constamment. Ces écoulements se caractérisent par une apparence très désordonnée et un comportement difficilement prévisible. Un agitateur produit des turbulences lorsque le mouvement du fluide est relativement intense devant les forces de viscosité (forces opposées au déplacement du fluide).

Puisque en conception préliminaire, on ne sait pas a priori les valeurs exactes ou on ne dispose pas des informations nécessaires pour les estimer ou calculer, les valeurs de ces variables peuvent être représentées par des ordres de grandeurs ou des valeurs qualitatives : faible, moyen et faible.

3.4 Éléments de la configuration d'un mélangeur

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, chicanes...). Tous ces choix demandent une certaine expertise. Un dispositif d'agitation complet est présenté en figure 6.12. Dans la cadre de l'exemple à traiter, on se limite aux parties : cuve, contre-pales (ou chicanes), dispositif thermique, mobile, arbre et moteur. Ces éléments sont détaillés par la suite. En effet, ces éléments influent sur les conditions de fonctionnement et la qualité du mélange et sont donc du ressort de l'ingénieur de génie des procédés (les autres éléments étant plus proches de la mécanique).

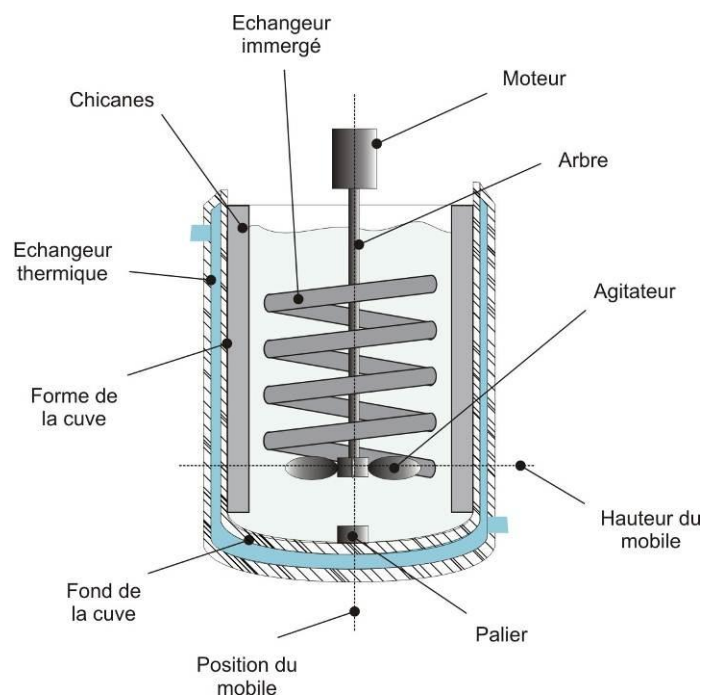


Figure 6.12 Le dispositif d'agitation

3.4.1 Les cuves

Selon l'application visée, la cuve peut prendre plusieurs formes, et avoir différents types de fonds (figure 6.13). La forme cylindrique est la plus répandue. Les fonds coniques et bombés favorisent la vidange et l'évacuation des liquides visqueux. Dans l'industrie, on privilégie souvent les formes carrés et cylindriques pour des raisons de gain de place. Le volume utile de la cuve correspond à une hauteur de liquide (mesurée depuis le point le plus bas) égale au diamètre de la cuve dans le cas d'une géométrie standard.

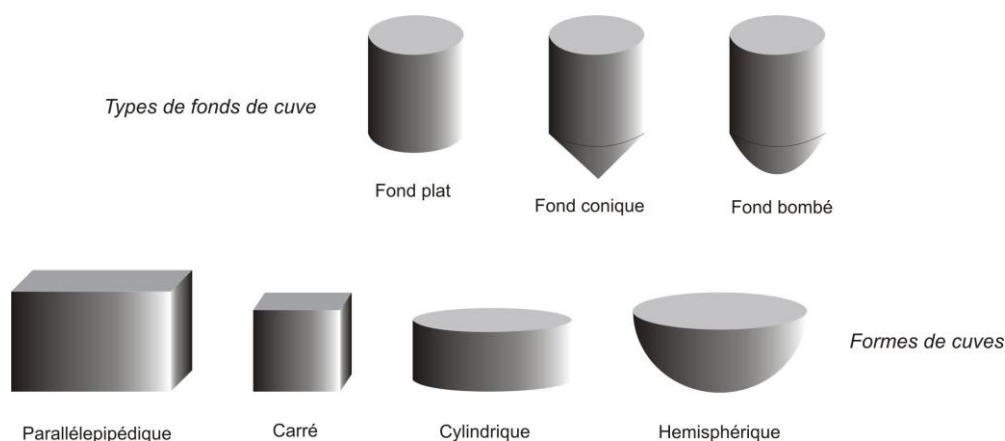


Figure 6.13 Les différentes formes de cuves

3.4.2 Les chicanes

Les chicanes sont un autre élément important dans la configuration d'un mélangeur. Elles servent notamment à éviter la formation de vortex. Ces derniers sont produits par la force centrifuge due à la rotation de l'agitateur. Les chicanes sont des plaques verticales fixées aux parois de la cuve. Elles sont indispensables pour obtenir un mélange efficace en régime d'écoulement turbulent. Avoir le même nombre de chicanes que de pales de l'agitateur permet que les efforts dus aux réactions lorsque les pales se trouvent en face des chicanes soient symétriques. La configuration standard prévoit quatre chicanes uniformément réparties sur le diamètre de la cuve. Si la cuve est de forme carré, elles ne sont pas nécessaires car les angles jouent le rôle de chicanes. Lorsqu'on mélange des liquides très visqueux, elles ne sont pas nécessaires car les vortex ne se forment quasiment pas. La figure 6.14 schématise les positions possibles pour les chicanes.

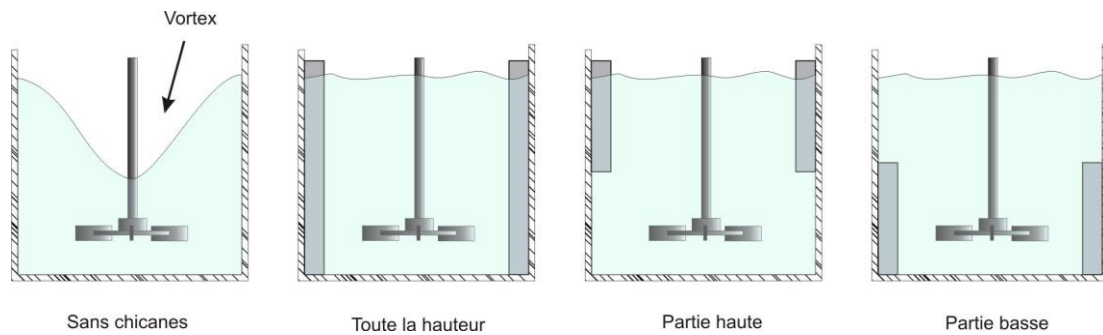


Figure 6.14 Les différentes positions des chicanes

3.4.3 Le dispositif thermique

Souvent l'opération de mélange est associée au chauffage ou au refroidissement du fluide dans la cuve. Il existe différents équipements possibles pour assurer ce transfert thermique. Les cuves destinées à ce genre d'opération sont généralement équipées soit d'un réseau de serpentins intérieurs, d'une double enveloppe, d'une demi-coquille ou d'une demi-coquille avec un serpentin comme illustré sur la figure 6.15. Les cuves destinées au transfert de chaleur peuvent généralement être équipées de tubes verticaux intérieurs.

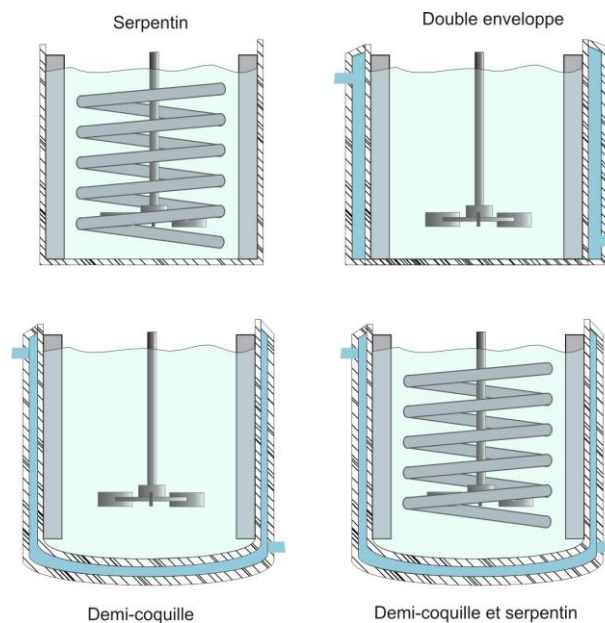


Figure 6.15 Cuves munies d'échangeurs de chaleur

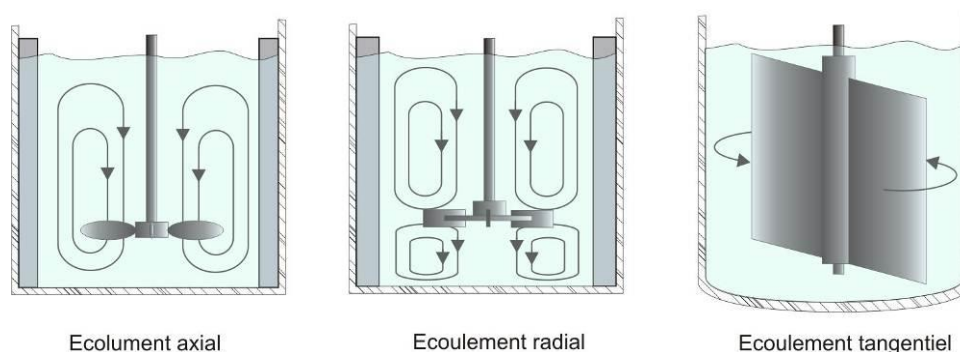
3.4.4 Les mobiles d'agitation

La plupart des opérations d'agitation et de mélange sont effectuées par un agitateur (ou mobile) tournant autour d'un axe à l'intérieur d'une cuve. Il existe quelques conditions à prendre en considération lors de l'utilisation d'un ou plusieurs mobiles d'agitation [XUE 06] :

- Pour homogénéiser des liquides miscibles avec ou sans différence de viscosité, de concentration ou de température, l'agitateur joue juste un rôle de générateur de mouvement.
- Pour mettre en contact des liquides non miscibles afin d'assurer le transfert de masse d'une phase vers l'autre, l'agitateur est un élément promoteur de la formation de gouttes de la phase dispersée dans la phase continue.
- Pour mettre en suspension des solides dans un liquide, l'agitateur a un double rôle : soulever les particules du fond de la cuve et les maintenir en suspension.
- Pour disperser un gaz dans un liquide, l'agitateur est responsable de la dispersion du gaz en créant des bulles de taille convenable et de la répartition de ces dernières dans tout le réacteur.

Cette liste non exhaustive démontre l'importance du choix du type d'agitateur comme un élément clé de la réussite du procédé. Il existe un grand nombre de modèles, chacun d'entre eux ayant une forme adaptée pour une application donnée. La figure 6.16 présente trois différents schémas d'écoulement en fonction du mobile d'agitation utilisé :

- Les agitateurs axiaux créent une circulation du fluide du haut vers le bas de la cuve,
- Les agitateurs radiaux développent un écoulement radial perpendiculaire à l'arbre de rotation. Le flux est expulsé vers les parois de la cuve et il se divise en créant deux boucles de circulation,
- Les agitateurs tangentiels déplacent les fluides en un seul bloc en rotation autour de l'axe du mobile d'agitation.



Source : [HER 10]

Figure 6.16 Différents schémas d'écoulement produits par les types de mobile d'agitation

Les mobiles d'agitation peuvent prendre plusieurs formes, le tableau 6.6 classe les plus usuels en fonction du type d'opération. Une liste plus détaillée est présentée en Annexe 5, le type de mobile étant une variable de notre modèle, cette liste représente son domaine de définition.

Nature de phase	Utilisation	Opération	Type de mobile d'agitation
Liquides miscibles	Homogénéisation Mise en circulation	Mélange Réaction chimique Dilution Neutralisation	Hélice, turbine ouverte pour fluides peu visqueux. Hélice double flux, ruban ou vis hélicoïdale pour fluides visqueux
Liquides immiscibles	Dispersion Emulsions	Réactions hétérogènes Extraction liquide/liquide	Hélice, turbine à disque, disque dilacérateur, Agitateurs rapides
Liquide-solide	Mise en suspension Maintien en suspension Délitage, Dispersion	Dissolution/Cristallisation Elutriation, Extraction liquide-solide, Lixiviation, Polymérisation Réaction avec catalyseur, Stockage de pulpes ou pates	Hélice double flux, turbine ouverte, disque dilacérateur
Liquide-gaz	Dispersion gaz-liquide	Absorption Désorption	Turbine à disque
	Chauffage ou refroidissement	Transfert de chaleur Aération Ozonation	Hélice pour fluides peu visqueux, ancre associé à un autre mobile pour fluides visqueux

Tableau 6.6 Quelques mobiles d'agitation en fonction de l'opération

3.4.4.1 La position du mobile d'agitation

La position du mobile d'agitation influence la structure hydrodynamique des écoulements turbulents. Par exemple une position non centrée du mobile dans la cuve peut être nécessaire pour de raisons d'encombrement. En général ce type de configuration réduit la création de vortex sans utiliser de chicanes. L'implantation latérale d'un agitateur est souvent utilisée dans de cuves à grand volume. Les figures 6.17 et 6.18 schématisent deux types de positions, une par rapport à la hauteur du mobile, et l'autre position pour déterminer l'axe du mobile en relation à la cuve.

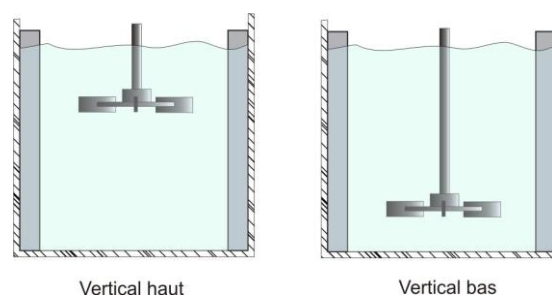


Figure 6.17 Hauteur du mobile d'agitation

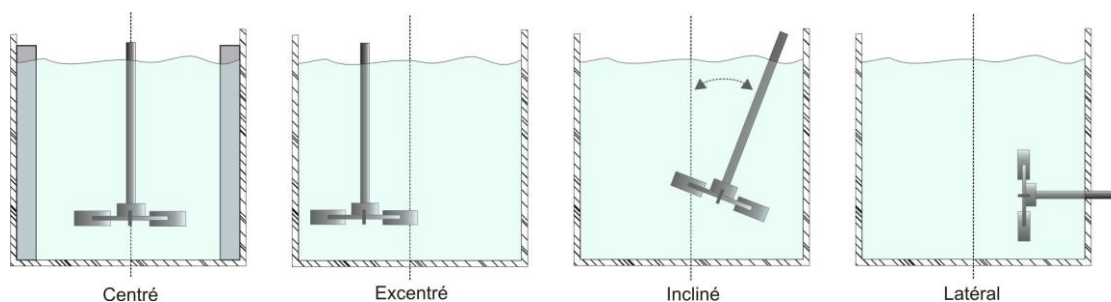


Figure 6.18 Positions du mobile d'agitation

3.4.5 L'entraînement

La mise en mouvement d'un fluide demande toujours une certaine quantité d'énergie. L'énergie est transmise au fluide au moyen du dispositif d'agitation et dissipée sous forme de chaleur. Il est important de pouvoir quantifier l'énergie dont l'opération aura besoin. La connaissance du besoin énergétique est un critère nécessaire pour le calcul de la motorisation mais aussi un élément essentiel pour la comparaison de l'efficacité entre mobiles.

Pour le problème d'exemple, la puissance nécessaire dépend directement de la puissance dans l'agitateur. Une équation permet de mettre en relation la puissance du moteur et la puissance de la turbine.

$$P_{\text{moteur}} \geq 2 \times P_{\text{turbine}}$$

$$\text{où } P_{\text{turbine}} = \rho \times \omega^3 \times pos \times (D_t)^5$$

ρ = la densité du produit à mélanger

ω = la vitesse de rotation

pos = la position de la turbine (elle dépend du type, si c'est radial, axial...)

D_t = le diamètre de la turbine

3.4.6 Les arbres

L'arbre d'agitation comprend deux parties liées : un arbre supérieur plein situé généralement à l'extérieur de la cuve, un arbre inférieur sur lequel est fixé le mobile d'agitation. Dans cette étude on se limite à l'arbre inférieur. Ce dernier est soit pendulaire soit guidé dans son extrémité basse par un palier de fond de cuve (ou crapaudine). On utilise ce guidage lorsque la flèche en bout d'arbre est jugée trop importante (arbres longs). L'avantage principale de ce système est la réduction des contraintes et donc du diamètre.

L'arbre peut être plein ou creux. A tenue mécanique équivalente, choisir un arbre creux permet de faire une économie de poids ce qui est déterminant pour le choix de l'agitateur. A noter que l'on s'est affranchit du problème des matériaux mais l'acier ou l'inox sont des matériaux classiquement utilisés dans les industries chimiques. Dans des applications particulières (milieu corrosif, normes de fabrication comme l'industrie agroalimentaires...) d'autres types de matériaux (Titane) ou des revêtements (Résine, Vitrification) peuvent être employés.

3.5 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 du cahier des charges fonctionnel, 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 6.19.

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

Figure 6.19 Données du problème PPC

Il faut ensuite préciser les variables et leurs domaines de définition. Dans le cas présent, l'architecture du système technique « Mélangeur » est composée de trois parties principales la Cuve, le Mobile et l'Entrainement. Chacune de ces parties englobe un certain nombre de variables accompagnées de leurs domaines associés. La figure 6.20, donne un aperçu des domaines de définition des variables, pour des raisons de lisibilité la liste n'est pas exhaustive.

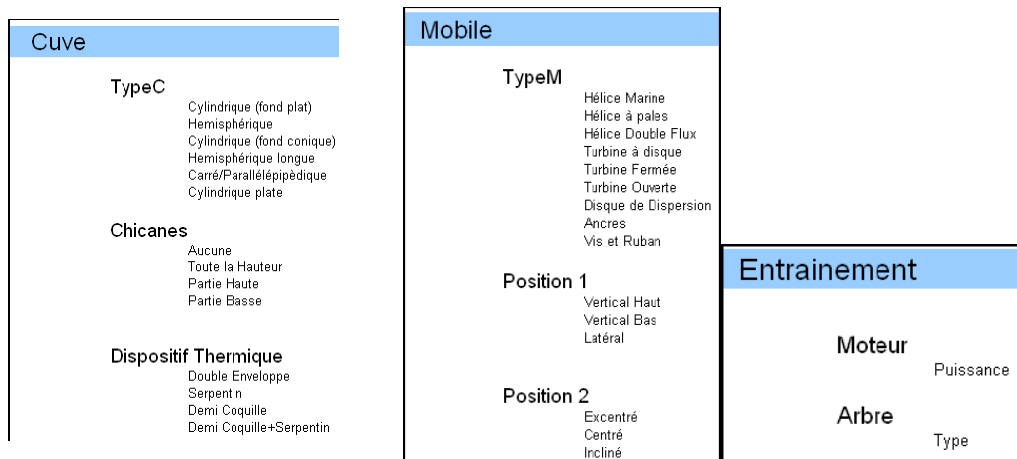


Figure 6.20 Variables et domaines du modèle générique

Pour compléter le modèle, les contraintes sont ajoutées, elles sont parfois enrichies d'un commentaire pouvant expliquer l'intérêt de cette contrainte. A cause de la taille du modèle, les contraintes sont exprimées en intention (expression synthétique de la restriction sur les domaines des variables). La figure 6.21 ci-dessous présente une extraction de quelques contraintes du modèle complet.

Choix du mobile d'agitation						
C(Etat; TO; Cisa; Turb; Pomp; viscositéL; TypeM)						
Etat	TO	Cisaill	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-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	Emulsion	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

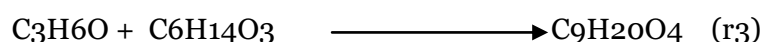
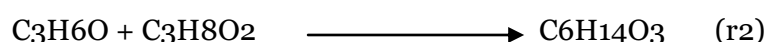
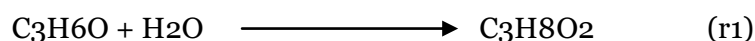
Présence de chicane				
C(Etat; Mouillabilité; Diamètre; Chicane)				
Etat	Mouillabilité	Diamètre	Chicane	Commentaire
Liquide-solide	Faible	Faible	Aucune	Présence d'un vortex qui a tendance à aspirer vers l'intérieur les particules de solide qui resteraient à la surface
Liquide-solide	Moyen	Moyen	Partie 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 ; Ciel; Solubilité; Chicane)				
Etat	Ciel	Solubilité	Chicane	Commentaire
Liquide-Gaz	oui	faible	aucune	Grâce au vortex réincorporatio
Liquide-Gaz	oui	Moyenne	aucune	
Liquide-Gaz	oui	Moyenne	Partie basse	
Liquide-Gaz	oui	Forte	Partie basse	

Figure 6.21 Exemples de contraintes du modèle Générique

3.6 Définition du problème pour la configuration d'un système d'agitation

Le propylène glycol est un produit largement utilisé dans les domaines de la chimie pharmaceutique et de la chimie de spécialité. Une voie de fabrication est de réaliser l'hydrolyse de 1,2 oxyde de propylène :



Cette synthèse fait intervenir 3 réactions consécutives compétitives conduisant à la formation de produits secondaires : di et tri glycols. Industriellement, la synthèse est réalisée avec un large excès d'eau de façon à favoriser la production de propylène glycol par rapport aux composés secondaires. Pour des raisons de solubilité, la réaction est conduite en présence de méthanol. Les paramètres cinétiques de la synthèse sont résumés dans le tableau suivant :

Réaction	Facteur pré-exponentiel (L/mol.s)	Energie D'activation (Kcal/Mol.K)	Ordre Réactionnel		Chaleur de Réaction (Kcal/Mol)
			A	B	
1	1,22 10 ⁸	18,0	1	1	-20,52
2	1,39 10 ¹⁰	21,1	1	1	-27,01
3	9,09 10 ¹¹	23,8	1	1	-25,81
Loi d'Arrhenius $r = k \cdot \exp\left(\frac{-E_a}{RT}\right) [A]^{O_a} [B]^{O_b}$					

Tableau 6.7 Paramètres cinétiques de la synthèse de 1,2 oxyde de propylène

Le réacteur est initialement rempli d'eau (avec 0,1% massique d'acide sulfurique utilisé comme catalyseur et négligé dans ce problème) et alimenté par un mélange équimoléculaire en oxyde de propylène et méthanol. La température initiale du mélange est de 20°C. Une contrainte de sécurité limitant l'augmentation de la température au sein du milieu réactionnel, est intégrée : température du milieu inférieure ou égale à 45°C (de plus, au-delà de 60°C certains constituants peuvent se dégrader). Une analyse de la cinétique de ces réactions montre que les réactions sont exothermiques et qu'une élévation de la température

du milieu est favorable à la création des produits (malheureusement plus favorable à la formation des produits secondaires).

3.7 Résolution du problème pour la configuration d'un système d'agitation

Suite à l'analyse des objectifs de l'opération de mélange, il est possible de décrire le problème. Il s'agit d'homogénéiser un mélange de liquides miscibles en vue d'une réaction chimique. Les fluides en présence sont Newtoniens, et connaissant la composition il est facile de pouvoir évaluer la viscosité et la masse volumique. Bien que ces propriétés varient entre le début et la fin de la réaction de par une composition différente du mélange (liée aux réactions), leurs ordres de grandeurs restent dans des intervalles de valeurs usuelles en génie des procédés. Quelques calculs préliminaires ont permis de déterminer l'intervalle de variation en fonction de la composition du mélange et de la température (utilisation du logiciel ProPhyPlus, avec Unifac comme modèle thermodynamique). Le but de l'opération de mélange est une homogénéisation des constituants en présence de façon à favoriser la réaction chimique, donc d'un point de vue qualitatif, les caractéristiques hydrodynamiques recherchées sont : un pompage fort, un cisaillement faible et une turbulence moyenne.

Le problème ainsi décrit, la remémoration est effectuée. Suite cette étape, le cas *source* sélectionné fournit le modèle générique β_{source} présenté dans la partie précédente. Ce cas contient également deux méthodes d'adaptation différentes. Elles sont rapidement éliminées par l'expert grâce au système d'aide à la décision et à la visualisation de ces méthodes. En effet, dans l'exemple proposé les deux méthodes éliminées comportent des contraintes trop spécifiques (contraintes pour imposer le type de cuve dans $MA^1(source)$, contrainte sur des conditions d'écoulement très particulières pour $MA^2(source)$ à cause de fluide non Newtonien). De plus la résolution du modèle β_{source} extrait (sans méthode d'adaptation), fournit un trop grand nombre de configurations possibles pour le système d'agitation requis (56 solutions candidates). A titre d'exemple, le domaine de la variable TypeMobile contient encore huit types d'agitateur potentiels : six mobiles à débit axial et deux à débit radial). Dans ces conditions, il a été décidé de créer une nouvelle méthode d'adaptation. La boucle avec l'expert est activée et la description du problème est progressivement améliorée.

La mise en place de palier de fond de cuve génère une friction dans les produits à mélanger et donc une pollution éventuelle du milieu. Or le produit désiré servant d'intermédiaire pour des utilisations pharmaceutiques (entre autres) cette pollution n'est pas

acceptable. Ainsi, un OA est spécifié pour enlever les valeurs « arbre creux avec palier et arbre plein avec palier » du domaine de la variable arbre.

Concernant le transfert thermique, une simulation a été réalisée afin de quantifier le flux énergétique à transférer. Dans les conditions opératoires désirées (Température initiale 20°C, Pression atmosphérique, composition connue), la simulation d'un réacteur adiabatique montre que la température du mélange atteint 70°C. Cette étude a permis d'acquérir des connaissances supplémentaires sur le cas d'étude et donc d'en déduire que le flux thermique pouvait être évacué avec un seul dispositif thermique (pas besoin de coupler un serpentin avec une double enveloppe par exemple). Dans l'exemple, l'introduction d'un dispositif thermique intrusif a pour conséquence de modifier l'hydrodynamique à l'intérieur de la cuve. Par conséquent, dans le domaine de définition du dispositif thermique, les valeurs contenant le terme serpentin ou tubes verticaux sont supprimées.

Suite à une analyse technico-économique, deux OA supplémentaires viennent ajouter des contraintes au modèle β_{source} . Tout d'abord, vu que le milieu est peu visqueux, les arbres creux répondent parfaitement aux conditions opératoires, ainsi ils permettent de réduire les coûts de fonctionnement du système. Une contrainte vient donc spécifier l'utilisation d'arbre creux. D'un point de vue hydrodynamique, l'agitateur devra posséder une capacité de pompage importante. Or pour une même puissance, le pompage est plus important avec une grande turbine tournant lentement qu'une petite turbine tournant rapidement. Mais à vitesse plus faible, une turbine de grand diamètre nécessite un couple plus important ce qui influe sur le système mécanique d'entraînement (partie mécanique non abordée dans le cadre de l'étude). En prenant en compte les problématiques dues au pompage important et à la réduction des coûts d'investissement liés à l'entraînement, une contrainte est ajoutée pour imposer le choix d'un agitateur axial.

La création de cette méthode d'adaptation, a nécessité des itérations. Durant ce processus de création la procédure de tests a été activée de façon à établir les OA problématiques lors de ces itérations mais également pour vérifier la cohérence et les conséquences des modifications successives. Cette procédure permet également de créer une méthode avec un minimum d'OA. Par exemple, la contrainte de la mise en œuvre d'un dispositif thermique à la paroi permet de limiter les configurations de cuve et ainsi d'éliminer les cuves carrées et parallélépipédiques (de plus rarement utilisées dans le domaine pharmaceutique) grâce à une contrainte du modèle β_{source} initial. Le tableau 6.6 dresse une synthèse des OA de la méthode d'adaptation créée.

OA	Description	Commentaire
<pre>domaine_effacer: arbre - {"creux_palier", "plein_palier"};</pre>	Efface les valeurs : « creux avec palier » et « plein avec palier » du domaine de la variable Arbre	Friction de palier causant de pollution dans le milieu
<pre>domaine_effacer: disp_therm - {"creux_palier", "plein_palier"};</pre>	Efface les valeurs : « serpentins » et « tubes verticaux » du domaine de la variable « Dispositif Thermique »	Température de 70°C au-delà de la limite imposée
<pre>constspec_ajouter: arbre = "creux";</pre>	Ajoute la contrainte spécifique pour la variable « Arbre »	Milieu peu visqueux
<pre>constspec_ajouter: agitateur = "axial";</pre>	Ajoute la contrainte spécifique pour la variable « Agitateur »	Pompage important, réduction de coûts liés à l'entraînement

Tableau 6.8 OA pour la configuration d'un dispositif agitateur

Le modèle β_{source} est modifié par la nouvelle méthode d'adaptation $MA^i(cible)$ et ensuite résolu par le solveur PPC. Au final ce modèle donne le choix entre 6 configurations envisageables qu'il faudra étudier en conception détaillée. Ces configurations sont présentées dans le tableau 6.9.

Type Cuve	Chicanes	Dispositif Thermique	Type Mobile	Position 1	Position 2	Arbre
Cylindrique fond plat	Non	Double enveloppe	Mobile à disque et pales inclinées	Vertical bas	Centré	Creux sans pales
Cylindrique fond plat	Toute la hauteur	Demi-coquille	Turbine à pales inclinées	Vertical bas	Centré	Creux sans pales
Cylindrique fond plat	Non	Demi-coquille	Hélice à pales à profil mince	Vertical bas	Centré	Creux sans pales
Cylindrique	Non	Demi-coquille	Hélice double flux	Vertical bas	Centré	Creux sans pales
Cylindrique	Non	Demi-coquille	Turbine à pales inclinées	Vertical bas	Centré	Creux sans pales
Cylindrique	Toute la hauteur	Demi-coquille	Turbine à pales inclinées	Vertical bas	Centré	Creux sans pales

Tableau 6.9 Les six configurations résultantes du dispositif mélangeur

La nouvelle méthode ainsi construite est sauvegardée dans la base en la liant avec le cas *source* remémoré et devient $MA^3(\text{source})$. Ce cas *source* contiendra alors 3 méthodes d'adaptation dont deux d'entre elles ($MA^2(\text{source})$) ainsi que la nouvelle $MA^3(\text{source})$) contiennent l'ajout d'une contrainte sur la prise en compte des aspects technico-économiques. Cette modification pourrait alors être appliquée directement dans le modèle générique et par conséquent cet OA pourrait être supprimé 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 les conclusions générales.

4 Conclusion

Pour les problèmes standards de conception, un système classique de RàPC (e.g. ReMSiProc) peut offrir des solutions rapidement, qui peuvent ensuite servir comme base à des analyses plus détaillées. L'outil RàP⁴C étend les performances du RàPC vers des problèmes plus complexes où les interactions avec les experts sont cruciales. L'architecture de RàP⁴C repose sur trois axes principaux :

- Une méthodologie pour l'adaptation de problèmes de conception.
- Un couplage avec la PPC, qui vise à formaliser les connaissances dans la base de cas sous la forme de PSC pour offrir des solutions à de problèmes complexes.
- Une interactivité avec les différents acteurs (i.e. expert ou utilisateur) afin d'acquérir ses connaissances et les rendre capitalisables.

Une application sur la conception, et plus particulièrement sur les choix technologiques pour un mélangeur industriel a été développée pour illustrer les possibilités de RàP⁴C dans la conception préliminaire d'opérations unitaires. Elle a démontré que l'approche d'adaptation présentée est pleinement applicable et capable d'être un outil d'aide à la conception dans ce domaine. L'étude de cas a également permis de valider la procédure de tests lors de l'établissement d'une nouvelle méthode d'adaptation. Grâce à la capitalisation des connaissances sur l'opération unitaire étudiée, le système permet ainsi de réduire le nombre de configurations envisageables pour le passage en conception détaillée.

Conclusions générales et perspectives

Au cours de ces dernières années, l'équipe Procédés et Systèmes Industriels du Laboratoire de Génie Chimique a prouvé l'efficacité de la Gestion des Connaissances pour la conception des procédés. Parmi les différentes approches en gestion des connaissances, le raisonnement à Partir de Cas (RàPC) a attiré particulièrement l'attention de par ses nombreux avantages (évoqués au chapitre 2). Dès lors des contributions ont été réalisées dans le but d'accroître les performances du RàPC : d'une part en proposant des méthodes génériques pour améliorer certaines étapes du cycle de RàPC, d'autre part en incluant des ontologies métiers pour le rendre plus applicable au domaine d'étude.

Cette thèse s'inscrit dans la continuité de ces travaux, et se focalise plus particulièrement sur l'étape d'adaptation : modification d'une solution remémorée pour proposer une solution appropriée au problème initial. En général, les problèmes de conception imposent les conditions d'adaptation les plus lourdes de par la quantité et la diversité des connaissances nécessaires. La motivation de cette thèse est de répondre deux questions : Comment réussir à adapter la solution d'un problème passé ? Et surtout comment générer, manipuler et réutiliser les connaissances nécessaires à cette étape pour arriver à une adaptation réussie ? Malgré sa complexité, l'adaptation de cas en RàPC reste considérée comme une étape cruciale mais difficile à mettre en œuvre car elle nécessite des connaissances supplémentaires (d'adaptation) qui posent des problèmes d'acquisition. Ainsi, une contribution a été développée afin de faciliter l'acquisition des connaissances auprès d'un expert métier.

Dans cette thèse l'accent a d'abord été mis sur les techniques de capitalisation des connaissances et surtout sur celles plus particulièrement adaptées à la conception en génie des procédés : le RàPC (chapitre 2) et la Programmation Par Contraintes (PPC) (chapitre 3). Pour ces chapitres des exemples illustrent l'application des approches dans le domaine d'étude. Le chapitre 4 présente les fondements et la problématique du processus d'adaptation. Il aborde le problème de l'acquisition des connaissances d'adaptation avec notamment les aspects liés à : la source de ces connaissances, leurs formulations, les méthodes utilisées pour les acquérir et enfin le moment de leurs acquisitions. L'approche proposée est détaillée dans le chapitre 5. Elle est basée sur le couplage de la PPC et du RàPC et sur une modification du cycle traditionnelle du RàPC afin de créer une boucle d'interaction

avec l'expert. Cette boucle a pour utilité d'acquérir les connaissances de l'expert durant un épisode de résolution et d'ajouter de l'interactivité au RàPC.

L'approche proposée est née de la volonté de réduire l'effort d'ingénierie pour l'acquisition et la formalisation des connaissances d'adaptation. Grâce au choix de définir la partie solution d'un cas sous la forme d'un PSC, l'ajout des connaissances supplémentaires est réalisé par modification du modèle via des opérateurs d'adaptations. La succession des opérateurs modifiant le modèle est rassemblée dans ce qu'on nomme une méthode d'adaptation. Ceci présuppose que l'adaptation peut se décomposer en actions élémentaires. Ces méthodes sont ensuite stockées dans la base de cas avec leur cas d'origine. Lors de la remémoration l'utilisateur choisit non seulement un cas source mais aussi une méthode d'adaptation. Une procédure de test de la méthode peut lui servir d'aide lorsqu'il n'arrive pas à identifier les opérateurs d'adaptation qui ne lui permettent pas d'atteindre une solution conforme à ses exigences. La procédure de tests et la modification des méthodes d'adaptation sont réalisées lors de l'activation de la boucle d'interaction, conduisant ainsi à une acquisition en ligne des connaissances (acquisition opportuniste).

Contributions

Les contributions de la thèse portent sur l'étape d'adaptation du RàPC, et plus particulièrement sur l'acquisition de connaissances d'adaptation. D'abord une modélisation des cas est proposée qui s'appuie sur le formalisme des PSC. Ce formalisme offre un cadre adéquat pour une application à des problèmes de conception car une fois les connaissances formalisées, une résolution indépendante du domaine d'application peut être réalisée. Cette formalisation permet une réutilisation des connaissances afin de supporter de futures conceptions. Puisque la conception est un processus dynamique, les modèles peuvent faire l'objet de modifications. Le formalisme de la PSC facilite ces manipulations et l'évolution des modèles (créés lors d'expériences antérieures). En formulant les cas comme des modèles de PSC on peut systématiser son traitement tout au long des étapes de RàPC mais aussi sur l'ensemble des évolutions du modèle.

Une autre contribution porte sur la proposition d'une stratégie d'acquisition de connaissances d'adaptation. Cette acquisition est conduite à partir des interactions avec l'utilisateur durant le processus d'adaptation. Cela réduit l'effort d'ingénierie de la connaissance par une sollicitation ponctuelle de l'expert lorsqu'il utilise le système pour résoudre un problème. Ainsi une boucle d'interaction est activée lorsqu'un échec d'adaptation

est rencontré et l'utilisateur peut corriger directement les causes de l'échec. Le système proposé est organisé et implémenté de telle sorte que l'utilisateur garde le contrôle des opérations durant l'adaptation. Avec l'intégration de la PPC des solutions sont calculées à chaque modification, ce qui permet à l'utilisateur de voir l'impact de ses décisions sur les solutions et cela ajoute de l'interactivité au processus de résolution. De plus, pour améliorer la compréhension des connaissances acquises, l'approche prend en compte les explications qu'un utilisateur peut éventuellement adjoindre aux modifications. Finalement les modifications sont stockées et mises à disposition sous la forme d'une méthode d'adaptation qui regroupe l'ensemble de modifications appelées Opérateurs d'adaptation.

Lorsqu'un cas possède plusieurs méthodes d'adaptation, un système d'aide à la décision est proposé afin de faciliter la sélection d'une méthode appropriée. Ce système porte sur des évaluations qu'un utilisateur effectue lorsqu'il a utilisé une méthode d'adaptation. En effet, l'utilisateur peut donner son avis et qualifier la performance de la méthode grâce à deux critères : la précision et la parcimonie. En fonction de ces critères, un front de Pareto est tracé afin d'aider au choix d'une méthode d'adaptation.

Toutes les contributions ont été concrétisées par la mise en œuvre de RàP⁴C, qui est un outil de raisonnement à partir de cas pour assister la conception dans le domaine du génie de procédés.

Perspectives

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 industriel crée un large panel d'options pour potentiellement améliorer les procédés. A l'heure actuelle, il existe des méthodes (et outils) pour la conception des procédés intensifiés, mais elles sont souvent focalisées sur une option spécifique. 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. Ce cadre s'appuiera sur les éléments développés par l'équipe autour du RàPC et notamment sur les contributions de cette étude. Cette perspective amènera à continuer le développement de l'approche selon trois axes.

Le premier cible la gestion des connaissances dans la base de cas. 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 basé sur les PSC. 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 remémoré 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 ? Et plus largement, il faudra se pencher sur la dernière étape du RàPC (non abordée dans l'outil actuel), avec notamment la maintenance des cas et de la base de cas.

Cette application va demander des évolutions nécessaires de l'outil, et plus particulièrement au niveau de la flexibilité de représentation des contraintes dans les modèles de PSC. En effet, dans le cadre de travail actuel, la représentation des connaissances est limitée à des contraintes analytiques dont la forme est une expression mathématique, une expression logique, un ensemble de combinaisons de valeurs autorisées. Mais en conception des procédés, les données à disposition prennent d'autres formes (notamment pour les aspects numériques) : abaques ou tableaux de valeurs, résultats expérimentaux (courbes, corrélations...), résultats de code de calcul... Il faudra donc intégrer ces structures de données sous forme de contraintes.

Le dernier axe se focalise sur le RàPC, en lui donnant une dimension collaborative. Que ce soit en intensification ou plus largement quel que soit l'objectif de conception, l'ingénieur procédé interagit avec des experts de différentes disciplines. En effet, le génie des procédés est une science d'intégration qui repose sur et utilise de multiples domaines et donc connaissances de base en vue de maîtriser les procédés dans leur totalité. Dans ces conditions, les connaissances et les cas peuvent être organisés en plusieurs sous-domaines : Thermique, Mécanique des fluides, Mécanique... Etant donné ce découpage des connaissances, l'adaptation et plus largement le processus complet de RàPC peut s'organiser en points de vue. Dans la communauté du RàPC, des recherches sont en cours sur ces RàPC multi points de vue, et il apparaît opportun de les appliquer au génie des procédés. Plus largement, dans une entreprise de nombreux projets ou situations font intervenir une équipe pluridisciplinaire de taille variable en fonction du projet ou de la situation : de quelques acteurs internes à l'entreprise à plusieurs personnes internes ou externes. Ce type de

collaboration peut se réaliser selon plusieurs échelles de temps en fonction de l'objectif de la collaboration : furtive dans le cas d'une gestion crise suite à un accident, à plus long terme pour la conception d'un procédé. Ces situations collaboratives présentent la particularité d'être dynamiques et contraintes dans le temps. Dans ces situations, la résolution du problème passe par des décisions des intervenants, plus ou moins indépendamment les uns des autres, mais qui doivent être mutualisées pour des raisons d'efficacité et de synergie des moyens. Pour prendre une décision, un intervenant doit considérer une situation dans sa globalité, car sa décision peut se concrétiser par une situation qui modifie l'environnement et provoquer des interactions. Pour cette perspective, il est envisagé de coupler le raisonnement à partir de cas à la notion de contexte pour former le « *raisonnement à partir de contexte* ». Un contexte peut se définir comme un ensemble de conditions et d'influences qui rendent une situation unique et compréhensible. L'élaboration d'un tel outil implique des verrous importants comme ; des contraintes liées au temps réel, des temps de décision limités, un système de décision des acteurs « léger » pour pouvoir accepter plusieurs intervenants.

En plus de sa vocation, première l'idée est d'exploiter un tel outil pour une autre finalité : la formation des acteurs (par exemples formation des opérateurs dans le cadre d'une application à la surveillance des procédés). Pour un tel outil, on envisage deux modes de fonctionnement :

1. Apprentissage de cas : dans ce mode, les experts simulent des prises de décision successives pour alimenter la base de cas. Chaque prise de décision est détaillée, argumentée et expliquée aux futurs utilisateurs.
2. Entraînement interactif : dans ce mode, les utilisateurs sélectionnent leurs actions respectives afin de s'entraîner. L'ensemble des décisions est mémorisé dans une « boîte noire » pour effectuer un retour d'expérience.

Bien qu'utile pour la conception, il est envisagé d'appliquer une telle approche à la surveillance des procédés : allant de la détection d'un dysfonctionnement à la résilience en passant par le diagnostic et la sécurisation de l'unité. En effet, dans cette application les temps de prise de décision sont très courts ce qui pose les conditions les plus défavorables à cause de l'accentuation des contraintes liées au temps réel.

Une nouvelle catégorie d'outils connus sous l'acronyme CAI (Computer-Aided Innovation) émerge parmi l'éventail des technologies assistées par ordinateur. L'évolution des CAI est une réponse aux demandes industrielles pour une plus grande fiabilité des nouveaux produits et procédés. L'objectif de ces outils en cours de développement est d'aider

les concepteurs dans leur performance créative, en vue de changements de paradigmes grâce à l'utilisation de cette nouvelle catégorie de logiciels. L'objectif de cette perspective à plus long terme est d'intégrer un tel outil dans les travaux menés au sein de l'équipe PSI portant sur les méthodes d'innovation avec notamment le couplage TRIZ-RàPC [COR 06] d'une part et par la prise en compte des contraintes environnementales [BAR 10] d'autre part.

Bibliographie

- [A&B 00] ARSHADI N. et BADIE K., « A Compositional Approach to Solution Adaptation in Case-Based Reasoning and its Application to Tutoring Library », Proceedings of the 8th German Workshop on Case-Based Reasoning (GWCBR'00), Germany, 2000.
- [A&K 08] AVRAMENKO Y. et KRASLAWSKI A., « Case Based Design, Applications in Process Engineering », Studies in Computational Intelligence, vol. 87, Springer, 2008.
- [A&P 94] AAMOD A. et PLAZA E., « Case-Based Reasoning: foundational issues, methodological variations, and systems approaches », AI Communications, IOS Press, vol. 7, pp. 39-59, 1994.
- [A&W 92] ACORN T. et WALDEN S., « SMART: Support Management Automated Reasoning Technology for Compaq Customer Service », Proceedings of the Innovative Applications of Artificial Intelligence Conference IAAI-92, 1992.
- [AAM 95] AAMOD A., « Knowledge acquisition and learning by experience: The role of case-specific knowledge », Machine learning and knowledge acquisition, Integrated approaches, Academic Press, pp. 197-245, 1995.
- [ABB 99] ABBASS H.A., WIGGEN G., LAKSHMANAN R. et MORTON B., « Constraint logic programming as a paradigm for heat exchanger network retrofit », The International Journal of Computers and Chemical Engineering, supplement volume of ESCAPE-9, pp. S127-S130, 1999.
- [AKE 05] AKEN V. J. E., « Valid knowledge for the professional design of large and complex design processes », Design Studies, vol. 26, n° 4, pp. 379-404, 2005.
- [ALL 94] ALLEN B.P., « Case based reasoning: business applications », Communications of the ACM, vol. 37, pp. 40-45, 1994.
- [AQU 04] AQUIN M. D., BRACHAIS S., LIEBER J. et NAPOLI A., « Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas », 12^{ème} Atelier de Raisonnement à Partir de Cas, Villetaneuse, France, 2004.
- [AQU 04] AQUIN M. D., BADRA F., LAFROGUE S., LIEBER J., NAPOLI A. et SZATHMARY L., « Case Base Mining for Adaptation Knowledge Acquisition », Proceedings of the 20th International Joint Conference on Artificial Intelligence IJCAI07, pp. 750-755, 2007.
- [ARC 01] ARCOS J. L., « T-Air: A case-based reasoning system for designing chemical absorption », Lecture Notes in Artificial Intelligence, vol. 2080, pp. 576-587, 2001.
- [AVR 04] AVRAMENKO Y., NYSTRÖM L. et KRASLAWSKI A., « Selection of internals for reactive distillation column: case-based reasoning approach », Computer Chemical Engineer, vol. 28, pp. 37-44, 2004.

- [AVR 04A] AVRAMENKO Y., KRASLAWSKI A. et MENSHUTINA N., « Decision supporting system for design of wastewater treatment », *Computer-Aided Chemical Engineering*, vol. 14, pp. 337-342, 2004.
- [B&C 07] BOUABDALLAH, R., CHIKH, A., « Adaptation des cas avec les réseaux de neurones », 15^{ème} Atelier de Raisonnement à Partir de Cas, pp. 111-118, Grenoble, France, 2007.
- [B&D 06] BENTEBIBEL R. et DESPRES S., « SAARA : Un Système d'Aide à l'Aménagement Routier Automatisé », 14^{ème} Atelier de Raisonnement à Partir de Cas, pp. 101-110, Bensaçon, France, 2006.
- [B&D 06A] BODIRSKY M. et DALMAU, V., « Datalog and constraints satisfaction with infinite templates », *Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science (STACS'06)*, vol. 3884, pp. 646–659, 2006.
- [B&G 06] BENHAMOU F. et GRANVILLIERS L., « Continuous and Interval Constraints », Rossi F., Van Beek P. et Walsh T., editors, *Handbook of constraint programming*, ch. 16, pp. 571-603, Elsevier, 2006.
- [B&M 01] BANDINI, S., MANZONI, S., « CBR adaptation for chemical formulation », *Lecture Notes in Computer Science*, vol. 2080, pp. 634-647, 2001.
- [BAD 07] BADRA F., LIEBER J. et NAPOLI A., « Extraction de connaissances d'adaptation par analyse de la base de cas », INRIA, Rapport préliminaire, 2007.
- [BAD 09A] BADRA F., CORDIER A. et LIEBER J., « Opportunistic Adaptation Knowledge Discovery », *Lecture Notes in Artificial Intelligence*, vol. 5650, pp. 60–74, 2009.
- [BAD 09B] BADRA F., « Extraction de connaissances d'adaptation en raisonnement à partir de cas », Thèse doctorale à l'Université Henri Poincaré - Nancy 1, France, 2009.
- [BAH 94] BAhLER D., DUPONT C. et BOWEN J., « An axiomatic approach that supports negotiated resolution of design conflicts in concurrent engineering », *Artificial Intelligence in Design*, pp. 363-379, 1994.
- [BAH 06] BAhLOUL D., « Une approche hybride de gestion des connaissances basée sur les ontologies : application aux incidents informatiques », Thèse doctorale à L'Institut National des Sciences Appliquées de Lyon, 2006.
- [BAI 11] BAILLEUX O., « Programmation par contraintes », Support de cours de Master 2 BD-IA et Image-IA, UFR Sciences et Techniques, Université de Bourgogne, France, 2011.
- [BAR 89] BAREISS R., « Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning », Academic Press, San Diego, CA, 1989.
- [BEN 94] BENHAMOU F., MCALLESTER D. A. et HENTENRYCK P. V., « CLP(Intervals) Revisited ». *Bruynooghe*, pp. 124–138, 1994.

-
- [BEN 10] BENELALLAM I., « Approches complètes pour la résolution des problèmes DisCSPs et DCOPs », Thèse doctorale à la Faculté des Sciences de Rabat, Maroc, 2010.
 - [BER 03] BERGMANN R., ALTHOFF K., BREEN S., GOKER M., MANAGO M. et TRAPHÖNER R., « Developing Industrial Case-Based Reasoning Applications – The INRECA Methodology », Lecture Notes in Artificial Intelligence, vol. 1612, 2003.
 - [BES 06] BESSIERE C., « Constraint Propagation », Handbook of Constraint Programming, Elsevier, vol. 2, ch. 3, pp. 29-83, 2006.
 - [BIE 97] BIEGLER L. T., GROSSMANN I. E. et WESTERBERG A.W., « Systematic Methods of Chemical Process Design », Prentice Hall, Upper Saddle River, 1997.
 - [BIN 10] BIN E., EMEK R., SHUREK G. et ZIV. A., « Using a constraint satisfaction formulation and solution techniques for random test program generation », IBM Systems Journal, vol. 41, n° 3, pp. 386–402, 2010.
 - [BOI 05] BOIVIN S., « Résolution d'un problème de satisfaction de contraintes pour l'ordonnancement d'une chaîne d'assemblage automobile », Mémoire de master à l'Université du Québec, Chicoutimi, Canada, 2005.
 - [BON 07] BONJOUR E., « Du retour d'expérience au développement des compétences », Dans Retour et Capitalisation d'expérience, outils et démarches, Bonjour J. et al. (Eds.), ch. 5, pp.137-178, 2007.
 - [BRO 08] BRODU N., « Query Sphere Indexing for Neighborhood Requests », Journal of Graphics Tools, vol. 13, n° 3, pp. 35-51, 2008.
 - [BUC 92] BUCKLEY M.J., FERTIG K.W. et SMITH D.E., « Design sheet: An Environment for facilitating flexible trade studies during conceptual design », American Institute for Aeronautics and Astronautics AIAA 92–1191, Aerospace Design Conference, Irvine, California, 1992.
 - [BUN 66] BUNGE, M., « Technology as applied science », Technology and Culture, vol. 7, n° 3, pp. 329–347, 1966.
 - [C&F 06] CORDIER A. et FUCHS B., « Apprendre à mieux adapter en Raisonnement à Partir de Cas », Dans Actes du 14^{ème} atelier de raisonnement à partir de cas (RàPC'06), Besançon, France, 2006.
 - [C&L 01] CORCHADO J. et LEES B., « Adaptation of cases for case-based forecasting with neural network support ». Soft Computing in Case-Based Reasoning, ch. 13, pp. 293–319, 2001.
 - [C&M 01] CUSSLER E.L. et MOGGRIDGE G.D., « Chemical Product Design », Cambridge Press, 2001.

- [CAR 83] CARBONELL J. G., « Learning by analogy : Formulating and generalizing plans from past experience », *Machine Learning, An Artificial Intelligence Approach*, ch. 5, pp. 137-161, Morgan Kaufmann, Inc., 1983.
- [CEN 04] COMITÉ EUROPÉEN DE NORMALISATION, CWA 14924-5:2004 E, « European Guide to good Practice in Knowledge Management – Part 5: KM Terminology », 2004.
- [CFM 06] CORDIER A., FUCHS B. et MILLE A., « De l'Ingénierie à l'Apprentissage des connaissances d'adaptation en Raisonnement à Partir de Cas », *Ingénierie des Connaissances, Semaine de la connaissance*, pp. 41-50, 2006.
- [CHA 02] CHAKRABARTI A., LANGDON P., LIU Y. et BLIGH T., « An approach to compositional synthesis of mechanical design concept using computers », *Engineering Design Synthesis*, Springer, Londres, pp. 179-197, 2002.
- [CHA 07] CHABER G., « Techniques d'intervalles pour la résolution de systèmes d'équations », Thèse doctorale à l'Université de Nice-Sophia Antipolis, Nice, France, 2007.
- [CHE 07] CHENOUIARD R., « Résolution par Satisfaction de Contraintes appliqué à l'aide à la décision en conception architecturale », Thèse doctorale à l'École Nationale Supérieure d'Arts et Métiers, Paris, France, 2007.
- [CHO 86] CHOURAQUI E., « Le raisonnement analogique : sa problématique, ses applications », *Actes des Journées Nationales sur l'Intelligence Artificielle, Aix-les-Bains*, pp. 107-117, CEPADUES-Editions, Toulouse, 1986.
- [CHO 09] CHONG Y., CHEN C. et LEONG F., « A heuristic based approach to conceptual design », *Research engineering design*, vol. 20, pp. 97-116, 2009.
- [COC 11] COCO, « Cape open to cape open simulation environment », 2011, <http://www.cocosimulator.org/>
- [COD 10] CODET DE BOISSE A., VAREILLES E., COUDERT T. et GENESTE L., « Couplage CSP et CBR : Premières Identifications des Modes de Couplage », 8th International Conference of Modeling and Simulation - MOSIM'10, Hammamet, Tunisia, 2010.
- [COO 05] COOPER R.G., « Product Leadership: Creating and Launching Superior New Products », 2nd Edition Basic Books Cambridge, 2005.
- [COR 06] CORTES ROBLES G., « Management de l'innovation technologique et des connaissances : synergie entre la théorie TRIZ et le Raisonnement à Partir de Cas. Application en génie des procédés et systèmes industriels », Thèse de doctorat à l'Institut National Polytechnique de Toulouse, ENSIACET-LGC, France, 2006.
- [COR 07] CORDIER A., FUCHS B., LIEBER J. et MILLE A., « Acquisition de connaissances du domaine d'un système de RàPC : une approche fondée sur l'analyse interactive des

- échecs d'adaptation --- le système FrakaS », Actes du 15^{ème} atelier de raisonnement à partir de cas, pp. 57-70, Grenoble, 2007.
- [COR 07A] CORDIER A., FUCHS B., LIEBER J. et MILLE A., « Acquisition interactive de connaissances d'adaptation intégrée aux sessions de raisonnement à partir de cas – Principes, architecture IAKA et prototype KAYAK », Actes du 15^{ème} atelier de raisonnement à partir de cas, pp. 71-84, Grenoble, 2007.
- [COR 08] CORDIER A., FUCHS B., LANA DE CARVALHO L., LIEBER J. et MILLE A., « Opportunistic Acquisition of Adaptation Knowledge and Cases — The IAKA Approach », K.-D. Althoff et al. (Eds.): ECCBR 2008, Lecture Notes in Artificial Intelligence, vol. 5239, pp. 150–164, 2008.
- [CZI 05] CZINER K., VIRKKI-HATAKKA T., HURME M. et TURUNEN I., « Evaluative approach for process development », Chemical Engineering Technologies, vol. 28, pp. 1490-1499, 2005.
- [D&G 02] DIAZ-AGUDO B. et GONZALEZ-CALERO P. A., « CBRONTO : A task/method ontology for CBR ». Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference (FLAIRS Conference), May 14-16, Pensacola Beach, Florida, USA, pp. 101–105, 2002.
- [D&P 00] DAVENPORT T. H. et PRUSAK L., « Working knowledge: How organizations manage what they know », Boston, MA: Harvard Business School Press, 2000.
- [DAQ 04] D'ACQUIN M., BRACHAIS S., LIEBER J. et NAPOLI A., « Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas.- Une approche fondée sur des techniques d'extraction de connaissances dans des bases de données », 12^{ème} atelier raisonnement à partir de cas, 2004.
- [DAR 05] DARSES F., « Contribution de l'ergonomie cognitive a la construction d'un modèle d'expertise des activités de conception de produits» Dans Ingénierie des Connaissances, Editions L'Harmattan, 2005.
- [DEC 90] DECHTER R. « Enhancement schemes for constraint processing: backjumping, learning, and cutset decomposition », Artificial Intelligence, vol. 41, n° 3, pp. 273–312, 1990.
- [DEL 00] DELOBEL F., « Résolution de contraintes réelles non-linéaires », Thèse de doctorat de l'Université de Nice, Sophia Antipolis, 2000.
- [DOU 88] DOUGLAS J. M., « Conceptual Design of Chemical Processes », McGraw-Hill, New York, 1988.

- [ENS 10] ENSIACET, « Génie des Procédés et Informatique », dans http://www.ensiacet.fr/Web_A7/ENSIA7_FR/FORMATION/INGENIEUR/GPI/gpi.shtml
- [ERS 12] ERSHOV M., AVRAMENKO Y., HÄKKINEM J., KRASLAWSKI, A. et BELOGLAZOV I., « Quotation Tool for Process Equipment » *Computer-Aided Chemical Engineering*, vol. 30, pp. 442–446, 2012.
- [EST 06] ESTEVEZ I., DUBOIS S., GARTISER N., RENAUD J. et CAILLAUD E., « Le raisonnement à partir de cas est-il utilisable pour l'aide à la conception inventive », 14^{ème} *Raisonnement à Partir de cas*, Besançon, pp. 123-129, 2006.
- [EUR 11] EURODECISION, « Programmation par contraintes », 2011, dans <http://www.eurodecision.fr/programmation-par-contraintes>
- [EVB 96] EVBUOMWAN N. F. O., SIVALOGANATHAN S. et JEBB A. A., « A survey of design philosophies, models, methods and systems », *Proceedings of the Institution of Mechanical Engineers*, pp. 301-320, London, 1996.
- [F&F 83] FINKELSTEIN L. et FINKELSTEIN A. C. W., « Review of design methodology », *IEEE Proceedings A*, vol. 130, n° 4, pp. 213-222, 1983.
- [F&S 03] FINNIE G. et SUN Z., « R5 model for case-based reasoning », *Knowledge Based Systems*, vol.16, pp. 59-65, 2003.
- [FAB 05] FABER R., JOCKENHÖVEL T., et TSATSARONIS, G., « Dynamic optimization with simulated annealing », *Computer Chemical Engineering*, vol. 29, pp. 273–290, 2005.
- [FAG 06] FAGES F., « Programmation logique et contraintes », *Encyclopédie de l'informatique et des systèmes d'information*, pp. 1038-1049, Vuibert, 2006.
- [FAR 03] FARKAS T., AVRAMENKO Y., KRASLAWSKI A., LELKES Z. et NYSTRÖM L., « Selection of MINLP model of distillation column synthesis by case-based reasoning », *Computer-Aided Chemical Engineering*, vol. 14, pp. 113-118, 2003.
- [FEI 63] FEILDEN G. B. R., « Engineering design, The Feilden's Report », Her Majesty's Stationery Office, London, 1963.
- [FID 06] FIDJELAND M. K., « Distributed Knowledge in Case-Based Reasoning – Knowledge Sharing and Reuse within the Semantic Web ». Thèse de maître, Norwegian University of Science and Technology, 2006.
- [FIS 78] FISHMAN G. S., « Concepts and Methods in Discrete Event Digital Simulation », John Wiley and Sons, 1973.
- [FRE 95] FREUDER E. C., « Using inference to reduce arc consistency computation », *Proceedings of the 14th International Joint Conference on Artificial Intelligence IJCAI'95*, pp. 592–598. Morgan Kaufmann Publishers, Inc, 1995.

-
- [FRO 94] FRON A., « Programmation par Contraintes », Addison-Wesley, France, 1994.
 - [FUC 95] FUCHS B., MILLE A., et CHIRON B., « Operator Decision Aiding by Adaptation of Supervision Strategies », *Case-Based Reasoning Research and Development*, vol. 1010, pp. 23-32, 1995.
 - [FUC 97] FUCHS B., « Représentation des connaissances pour le raisonnement à partir de cas : le système ROCADE », Thèse de l'Université de Saint Etienne, 1997.
 - [FUC 00] FUCHS B., LIEBER J., MILLE A. et NAPOLI A. « An algorithm for Adaptation in Case-Based Reasoning », *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000)*, Berlin, Germany, pp. 45–49, 2000.
 - [G&B 06] GRANVILLIERS L. et BENHAMOU F., « Algorithm 852: RealPaver: An Interval Solver Using Constraint Satisfaction Techniques », *ACM Transactions on Mathematical Software*, vol. 32, n° 1, pp. 138–156, 2006.
 - [G&F 03] GELLE E. M. et FALTINGS B., « Solving Mixed and Conditional Constraint Satisfaction Problems », *Constraints*, vol. 8, n° 2, pp. 107-141, 2003.
 - [G&K 91] GOEL A. K. et KOLODNER J. L., « Towards a case-based tool for aiding conceptual design problem solving », *Proceedings DARPA Case-Based Reasoning Workshop*, pp. 109-120, 1991.
 - [G&P 06] GALUSHKA M. et PATTERSON D., « Intelligent index selection for case based reasoning », *Knowledge Based Systems*, vol.19, pp. 625-638, 2006.
 - [G&S 95] GELLE E. et SMITH I., « Dynamic Constraint Satisfaction with Conflict Management in Design », *OCS'95: Workshop on Over Constrained Systems CP'95*, 1995.
 - [GAS 79] GASCHNIG J., « Performance measurement and analysis of certain search algorithms », *Technical Report CMU-CS-79-124*, Carnegie-Mellon University, 1979.
 - [GEL 98] GELLE E., « On the generation of locally consistent solution spaces in mixed dynamic constraint problems », Thèse de doctorat à l'Ecole Polytechnique Fédérale de Lausanne, Suisse, 1998.
 - [GEL 00] GELLE E., FALTINGS B., CLEMENT D. et SMITH I. « Constraint Satisfaction Methods for Applications in Engineering », *Engineering with computers*, vol.16, pp. 81-95, Springer, London, 2000.
 - [GIN 93] GINSBERG M., « Dynamic backtracking », *Journal of Artificial Intelligence Research*, vol.1, pp. 25-46, 1993.
 - [GOE 97] GOEL A. K., BHATTA S. R. et STROULIA E., « KRITIK : An Early Case Based Design System », *Issues and Applications of Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, Mahwah, NJ, pp. 87-132, 1997.

- [GOK 98] GÖKER M. H., « Attribute types and adaptation of technical objects », Proceedings of the 6th German Workshop on Case-Based Reasoning (GWCBR-98), Universität Rostok, Berlin, vol. 7, pp. 169–178, 1998.
- [GUE 03] GUERET C., PRINS C. et SEVAUX M., « Programmation Linéaire – 65 problèmes d’optimisation modélisés et résolus avec Visual Xpress », Editions Eyrolles, France, 2003.
- [GUI 06] GUILLER F., PHILIPPE V., PHILIPPE J. et FOLLUT D., « ATHANOR : Une approche de gestion de connaissances procédurales pour la maintenance de systèmes complexes », Revue des Nouvelles Technologies de l’Information, Systèmes d’Information pour l’aide à la Décision en Ingénierie Système. Cépadués-Éditions, Toulouse, France, p. 21, 2006.
- [GRA 00] GRAY P., « Knowledge management overview », Center for Research on Information Technology and Organizations, University of California, U.S.A., 2000.
- [GRA 05] GRANVILLIERS L. et SORIN V., « Elisa Version 1.0.4 », 2005, dans <http://sourceforge.net/projects/elisa/>
- [GRO 02] GROSSMAN I. E., « Review of non-linear mixed integer and disjunctive programming techniques », Optimization and Engineering, U.S.A., 2002.
- [H&G 02] HARJUNKOSKI I. et GROSSMANN I. E., « Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods », Computers and Chemical Engineering, vol. 26, pp. 1533–1552, 2002.
- [H&H 92] HENNESSY D. H. et HINCKLE D., « Applying Case-Based Reasoning to Autoclave Loading », IEEE Expert, vol. 7, n°5, pp. 21-26, 1992.
- [H&K 96] HANNEY K. et KEANE M.T., « Learning Adaptation Rules From a Case-Base », I. Smith & B. Faltings, Eds., Advances in Case-Based Reasoning, Third European Workshop, EWCBR’96, Lecture Notes in Artificial Intelligence, vol. 1168, pp. 179-192, 1996.
- [H&K 97] HANNEY K. et KEANE M.T., « The Adaptation Knowledge Bottleneck: How to Unblock it By Learning From Cases », Proceedings of the 2nd International Conference on CBR, pp. 359–370, 1997.
- [H&V 95] HAIGH, K. et VELOSO, M., « Route planning by analogy », In Proceedings of the 1st International Conference on Case Based Reasoning, pp. 169-180, Portugal, 1995.
- [H&W 95] HANKS S. et WELD D. S., « A Domain-Independent Algorithm for Plan Adaptation », Journal on Artificial Intelligence Research, vol. 2, pp. 319-360, 1995.
- [HAD 02] HADJ-HAMOU H., « Contribution à la conception de produits à forte diversité et de leur chaîne logistique : une approche par contraintes », Thèse de doctorat à l’Institut National Polytechnique de Toulouse, France, 2002.

-
- [HAM 89] HAMMOND K. J., « Case-Based Planning », Academic Press, San Diego, CA, 1989.
 - [HAN 95] HANNEY K., KEANE M. T., SMYTH B. et CUNNINGHAM P., « Systems, Tasks and Adaptation Knowledge: Revealing Some Revealing Dependencies », *Lecture Notes in Computer Science*, vol. 1010, pp. 461-470, 1995.
 - [HAN 97] HANNEY K., « Learning Adaptation Rules from Cases ». Thèse de maître, Trinity College, Dublin, 1997.
 - [HAO 08] HAOUCHINE K., CHEBEL-MORELLO B. et ZERHOUNI N., « Algorithme d'Adaptation pour le Diagnostic Technique », *Mémoires du 16^{ème} Atelier de Raisonnement à Partir de Cas*, Nancy, France, 2008.
 - [HAR 95] HAROUD D., BOULANGER S., GELLE E. et SMITH I., « Management of conflict for preliminary engineering design tasks », *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 9, pp. 313-323, 1995.
 - [HEI 98] HEIKKILÄ A.-M., KOIRANEN T. et HURME, M., « Application of case-based reasoning to safety evaluation of process configuration », *HAZARDS XIV, Institution of Chemical Engineers Symposium Series*, n° 144, IChemE, Rugby, pp. 461-473, 1998.
 - [HEN 99] VAN HENTENRYCK P., « The OPL Optimization Programming Language », Cambridge, MA, MIT Press, 1999.
 - [HER 10] HERMAN C., « Contribution à l'étude de la cristallisation, par refroidissement en cuve agitée, de substances d'intérêt pharmaceutique présentant un polymorphisme cristallin », Thèse de doctorat à l'Université Libre de Bruxelles, Belgique, 2010.
 - [HIN 88] HINRICHS T. R., « Towards an Architecture for Open-World Problem Solving », *Proceedings of the 1988 DARPA Workshop on Case-Based Reasoning*, pp. 182-189, Clearwater Beach, Florida, USA, 1988.
 - [HIN 92] HINRICHS T. R., « Problem Solving in OpenWorlds: A case study in Design », Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.
 - [HOK 06] HO KON TIAT V., « Aide à la décision pour la conception préliminaire de procédés d'évaporation flash », Thèse de doctorat à l'Université de Bordeaux I, France, 2006.
 - [HUA 96] HUA K., FALTINGS B. et SMITH I., « CADRE: case-based geometric design », *Artificial Intelligence in Engineering*, vol. 10, pp. 171-183, 1996.
 - [HUN 95] HUNT J., « Evolutionary case based design », Watson (Ed.), *Progress in Case Based Reasoning, Lecture Notes in Artificial Intelligence*, vol. 1020, pp. 17-31, 1995.
 - [HYV 94] HYVONEN E., « Constraint Reasoning Based on Interval Arithmetic: The Tolerance Propagation Approach », Freuder E. C. et Mackworth A. K. (Ed.), *Constraint-Based Reasoning*, The MIT Press, ch. 3, pp. 71-112, 1994.

- [IMT 00] IMTR, « Integrated Manufacturing Technology Roadmapping Initiative, .Modeling & Simulation », IMTI, Inc, 2000, dans <http://www.imti21.org/>
- [INA 01] INAKOSHI H., OKAMOTO S., OHTA Y. et YUGAMI N., « Effective Decision Support for Product Configuration by Using CBR », Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR), Workshop Casebased Reasoning in Electronic Commerce, Vancouver, Canada, 2001.
- [J&B 96] JUSSIEN N. et BOIZUMAULT P., « CSP dynamiques et Relaxation de Contraintes », 2^{ème} Conférence Nationale sur la résolution pratique de Problèmes NP-Complets (CNPC'96), pp. 135-149, Dijon, France, 1996.
- [J&G 01] JAIN V. et GROSSMANN I. E., « Algorithms for hybrid MILP/CP model for a class of optimization problems », INFORMS Journal in Computing, vol. 13, pp. 258–276, 2001
- [JAR 01] JARMULAK J., CRAW S., ROWE R., « Using case-base data to learn adaptation knowledge for design », Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), Seattle,WA, Morgan Kaufmann, San Francisco, pp. 1011–1016, 2001.
- [K&H 92] KAMBHAMPATI S. et HENDLER J., « A validation structure based theory of plan modification and reuse », Artificial Intelligence, vol. 55, pp. 193-258, 1992.
- [K&H 97] KOIRANEN T. et HURME M., « Case-based reasoning applications in process equipment selection and design », 6th Scandinavian Conference of Artificial Intelligence SCAI'97: Research announcements, Publication C-1997-49, University of Helsinki, Department of Computer Science, pp. 28-37, 1997.
- [K&H 01] KHAN A. S. et HOFFMANN A., « Acquiring Adaptation Knowledge for CBR with MIKAS », Proceedings of the 14th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence, pp. 201-212, 2001
- [KAS 95] KASIF S., SALZBERG S., WALTZ D., RACHLIN J. et AHA D., « Towards a Framework for Memory-Based Reasoning », NECI Technical Report, pp. 95-132, 1995.
- [KHA 00] KHALIL, M.S., « Evolutionary methods in chemical engineering », Plant Design Report Series, n° 64, Helsinki University of Technology, Espoo, 2000.
- [KIN 99] KING J. M. P, BAÑARES-ALCANTARA R. et ZAINUDDIN A. M, « Minimising environmental impact using CBR: An azeotropic distillation case study », Environmental Modelling & Software, vol. 14, pp. 395-366, 1999.
- [KOL 92] KOLODNER J. L., « An introduction to case based reasoning », Artificial Intelligence Review, vol. 6, n° 1, pp. 3–34, 1992.
- [KOL 93] KOLODNER J. L., « Case-based Reasoning », Morgan Kaufmann, San Mateo, 1993.

-
- [KOT 88] KOTON P., « Reasoning about evidence in causal explanations », In: Kolodner, J. (ed.): 1st Workshop on CBR. Morgan Kaufmann, San Mateo, pp. 260-270, 1988.
 - [KRA 95] KRASLAWSKI A., KOIRANEN T. et NYSTRÖM L., « Case-based reasoning system for mixing equipment selection », *Computer Chemical Engineering*, vol. 19, suppl. S821-S826, 1995.
 - [L&N 96] LIEBER J. et NAPOLI A., « Using classification in case based planning », *Proceedings of ECAI 96*, pp. 132-136, 1996.
 - [L&P 07] LEAKE D. et POWELL J., « Mining Large-Scale Knowledge Sources for Case Adaptation Knowledge », *Lecture Notes in Artificial Intelligence*, vol. 4626, pp. 209–223, 2007.
 - [L&W 98] LEAKE D. et WILSON D. C., « Categorizing case-base maintenance: dimensions and directions » *Proceedings of the Fourth European Workshop of Case-Based Reasoning*, Dublin, Ireland, pp. 196–207, 1998.
 - [L&Z 99] LEWKOWICZ M., ZACKLAD M., « MEMO-net, un collecticiel utilisant la méthode de résolution de problème DIPA pour la capitalisation et la gestion des connaissances dans les projets de conception », *IC'99*, Palaiseau, pp. 119-128, 1999.
 - [LEA 95] LEAKE D., KINLEY A. et WILSON D., « Learning to Improve Case Adaptation by Retrospective Reasoning and CBR », *Proceedings of the 1st International Conference on CBR*, Sesimbra, Portugal, 1995
 - [LEA 96] LEAKE D., « Case-Based Reasoning: Experiences, Lessons, and Future Directions », Menlo Park AAAI Press/MIT Press, 1996.
 - [LEA 96A] LEAKE D. B., KINLEY A., WILSON D., « Acquiring case-adaptation knowledge: a hybrid approach », *Proceedings of the 13th National Conference on Artificial Intelligence*, Menlo Park, CA, AAAI Press, Menlo Park, CA, pp. 684–689, 1996.
 - [LEA 97] LEAKE D. B., KINLEY A., WILSON D., « Case Based Assessment: estimating adaptability from experience », *Fourteenth National Conference on Artificial Intelligence*, AAAI press, Menlo Park, CA, pp. 674-679, 1997.
 - [LEC 03A] LECOUTRE C., BOUSSEMART F., et HEMER F., « De AC3 à AC7 », 12^{ème} JFPLC'03, pp. 267-280, 2003
 - [LEC 03B] LECOUTRE C., BOUSSEMART F. et HEMER F., « Au cœur de la consistance d'arc », 9^{ème} JNPC'03, pp. 233-247, 2003.
 - [LEE 03] LEE M., « A study of an automatic learning model of adaptation knowledge for case base reasoning », *Information Sciences Journal*, vol. 155, pp. 61-78, 2003.
 - [LHO 93] LHOMME O., « Consistency techniques for numeric CSPs ». *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI 1993*, pp. 232-238, 1993.

- [LIE 01] LIEBER J., BEY P., BOISSON F., BRESSON B., FALZON P., LESUR A., NAPOLI A., RIOS M., SAUVAGNAC C., « Acquisition et modélisation de connaissances d'adaptation, une étude pour le traitement du cancer du sein, Journée Ingénierie des connaissances », IC'2001, Presses Universitaires de Grenoble, Grenoble France, pp. 409-426, 2001.
- [LIE 02] LIEBER J., « Recopier c'est déjà adapter : six types d'adaptation par copie », 10^{ème} Séminaire Français de Raisonnement à Partir de Cas - RàPC'2002, pp. 11-21, Paris, France, 2002.
- [LIE 04] LIEBER J., D'AQUIN M., BRACHAIS S et NAPOLI A., « Une étude comparative de quelques travaux sur l'acquisition de connaissances d'adaptation pour le raisonnement à partir de cas », Actes du 12^{ème} atelier de raisonnement à partir de cas, 2004.
- [LIE 06] LIEBER J., « A Definition and a Formalization of Conservative Adaptation for Knowledge-Intensive Case-Based Reasoning – Application to Decision Support in Oncology (A Preliminary Report) », Rapport de recherche, LORIA, 2006.
- [LIE 08] LIEBER J., « Contributions à la conception de systèmes de raisonnement à partir de cas », Habilitation à Diriger des Recherches, Université Poincaré, Nancy, 2008.
- [LIT 02] LITTLE J., GEBRUERS C., BRIDGE D. et FREUDER E., « Capturing Constraint Programming Experience: A Case-Based Approach », Cork Constraint Computation Centre, University College, Cork, Ireland, 2002.
- [LIX 02] LI X.N., RONG B.G. et KRASLAWSKI A., « Synthesis of reactor/separators networks by the conflict-based analysis approach », Computer-Aided Chemical Engineering, vol. 10, pp. 241–246, 2002.
- [LOP 03] LÓPEZ B., « Holiday scheduling for city visitors », Information and Communication Technologies in Tourism, Springer Computer Science, 2003.
- [LOP 05] LÓPEZ-ARÉVALO I., RODRÍGUEZ-MARTÍNEZ A., ARANTZA ALDEA, BAÑARES-ALCÁNTARA R. et JIMÉNEZ L., « Redesign Support Framework based on Hierarchical Multiple Models », Proceedings of the International Joint Conference on Artificial Intelligence IJCAI 2005, pp. 1570-1571, 2005.
- [LOP 06] 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., « Retrieval, reuse, revision and retention in case based reasoning », The Knowledge Engineering Review, vol. 20, n° 3, pp. 215-240, 2006.
- [LOP 07] LÓPEZ-ARÉVALO I., RODRÍGUEZ-MARTÍNEZ A., ARANTZA ALDEA, BAÑARES-ALCÁNTARA R. et JIMÉNEZ L., « Generation of process alternatives using abstract

- models and case based reasoning », *Computer Chemical Engineering*, vol. 31, pp. 902–918, 2007.
- [LUC 84] LUCKMAN, J., « An approach to the management of design, In *Developments in design methodology* », Ed. N. Cross, John Wiley, pp. 83-97, London, 1984.
 - [M&C 08] MUÑOZ-AVILA, H., COX, M. T., « Case-Based Plan Adaptation: An Analysis and Review », *Intelligent Systems, IEEE*, vol. 23, issue 4, pp. 75-81, 2008.
 - [M&G 04] MARAVELIAS C. T. et GROSSMANN I. E., « A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants », *Computers and Chemical Engineering*, vol. 28, pp. 1921–1949, 2004.
 - [M&H 95] MUNOZ H. et HUELLEN J., « Retrieving cases in structured domains by using goal dependencies », *Proceedings of the 1st International Conference on Case Based Reasoning*, pp. 241-252, Sesimbra, Portugal, 1995.
 - [M&P 97] MAHER M. L. et PU P., « Introduction to the Issues and Applications of Case-Based Reasoning in Design », LEA Publishers, ch. 1, pp. 1-9, London, 1997.
 - [M&T 94] MYLLYMÄKI, P. et TIRRI, H., « Massively parallel case-based reasoning with probabilistic similarity metrics », *Topics in Case-Based Reasoning, Lecture Notes in Artificial Intelligence*, vol. 837, pp. 144-154, 1994.
 - [M&W 96] MUNOZ H. et WEBERSKIRCH F., « Planning for manufacturing workpieces by storing, indexing and replaying planning decisions », *Proceeding of the 3rd International Conference on AI Planning Systems (AIPS-96)*, AAAI-Press, 1996.
 - [M&Z 93] MAHER M. L. et ZANG D. M., « Cadsyn: A case-based design process model, In *Process Model* », *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 7, n° 2, pp. 97-110, 1993.
 - [MAC 77] MACKWORTH A. K. « Consistency in network of relations », *Artificial Intelligence*, vol. 8, pp. 99-118, 1977.
 - [MAH 95] MAHER M. L., BALACHANDRAN M. B. et ZHANG D. M., « Case-Based Reasoning in Design », LEA Publishers, London, 1995.
 - [MAR 96] MARK W., SIMOUDIS E. et HINKLE D., « Case-Based Reasoning : Expectations and Results » *Case-Based Reasoning, Experiences, Lessons & Future Directions*, Edited by Leake D., ch. 14, pp. 269-294, 1996.
 - [MAR 02] MARLING C., SQALLI M., RISSLAND E., MUNOZ-AVILA H. et AHA D., « Case-Based Reasoning Integrations » *Artificial Intelligence Magazine, AAAI*, vol. 23, n° 1, pp. 69-86, 2002.
 - [MAR 06] MARLING C., RISSLAND E. et AAMODT A., « Integrations with case-based reasoning » *The Knowledge Engineering Review*, vol. 20, n° 3, pp. 241-245, 2006.

- [MED 09] MEDJDOUB B., « Constraint-based adaptation for complex space configuration in building services », Special Issue Building Information Modeling Applications, Challenges and Future Directions, Journal of Information Technology in Construction (ITcon), vol. 14, pp. 724-735, 2009
- [MEN 05] MENDEZ C. A., CERDA J. et GROSSMANN I. E., « State-of-the-art review of optimization methods for short-term scheduling of batch processes », ABB Corporate Research Center, Ladenburg, Germany, 2005.
- [MEY 04] MEYER X., « Habilitation à Diriger des Recherches », Institut National Polytechnique de Toulouse, 2004.
- [MIL 96] MILLE A., FUCHS B. et HERBEAUX O., « A unifying framework for Adaptation in Case-Based Reasoning ». Proceedings of the ECAI'96 Workshop: Adaptation in Case-Based Reasoning, pp. 22-28, 1996.
- [MIN 92] MINTON S., JOHNSTON M.D., PHILIPS A.B. et LAIRD P., « A heuristic repair method for constraint-satisfaction and scheduling problems », Artificial Intelligence, vol. 58, pp. 161-205, 1992.
- [MON 08] MONTICOLO D., « Une approche organisationnelle pour la conception d'un système de gestion des connaissances fondé sur le paradigme agent », Thèse de doctorat à l'Université de Technologie de Belfort-Motbéliard, France, 2008.
- [MOO 66] MOORE R. E., « Interval Analysis », Prentice Hall, 1966.
- [MOR 12] MORALES-MENDOZA L. F., AZZARO-PANTEL C., BELAUD J. P., PIBOULEAU L. et DOMENECH S., « An integrated approach combining process simulation and life cycle assessment for ecoefficient process design », Computer-Aided Chemical Engineering, vol. 30, pp. 142-146, 2012.
- [N&L 08A] NEGNY S. et LE LANN J. M., « Case Based Reasoning for Chemical Engineering Design », Chemical Engineering Research and Design, vol. 86, n° 6, pp 648-658, 2008.
- [N&L 08B] NEGNY S. et LE LANN J. M., « Acceleration of the Retrieval of past experiences in Case Based Reasoning: application for preliminary design in Chemical Engineering », Computer Aided Chemical Engineering, vol. 25, pp 1009-1014, 2008.
- [N&T 97] NONAKA I. et TAKEUCHI H., « La connaissance créatrice. La dynamique de l'entreprise apprenante » traduit par De Boeck Université, 1997.
- [N&V 96] NETTEN B. D. et VINGERHOEDS R. A., « Adaptation for conceptual design in ADOCS », Proceedings of the ECAI'96 Workshop on Case Adaptation, Budapest, Hungary, 1996.
- [NAV 88] NAVINCHANDRA D., « Case-based reasoning in CYCLOPS, a design problem solver », Proceedings of the Case-Based Reasoning Workshop (DARPA). Morgan-Kaufmann Publishers, Inc., San Mateo, CA, 1988.

-
- [NEA 05] NEAGU N., « Constraint Satisfaction Techniques for Agent-Based Reasoning », Whitestein Series in Software Agent Technologies, Birkhäuser Verlag, Berlin, 2005.
 - [NEG 10] NEGY S., RIESCO H. et LE LANN J. M., « Effective retrieval and new indexing method for case based reasoning: Application in chemical process design », Engineering Applications of Artificial Intelligence, vol. 23, pp 880–894, 2010.
 - [NEU 90] NEUMAIER A., « Interval Methods for System Equations », Encyclopedia of Mathematics and its Applications, vol. 37, 1990.
 - [OLI 09] OLIVIER MAGET N., NEGY S., HETREUX G. et LE LANN J. M., « Fault diagnosis and process monitoring through model based CBR » Computer Aided Chemical Engineering, vol. 26, pp. 345-350, 2009.
 - [OUD 06] VAN OUDERNHOVE DE SAINT GÉRY T., « Contribution à l'élaboration d'un formalisme gérant la pertinence pour les problèmes d'aide à la conception à base de contraintes », Thèse de doctorat à l'Institut National Polytechnique de Toulouse, France, 2006.
 - [P&B 96] PAHL G. et BEITZ W., « Engineering Design: A systematic approach », Springer-Verlag, 2nd édition, Berlin, 1996.
 - [P&G 07] PRALUS M. et GENESTE L., « Recherche et adaptation d'expériences structures, imprécises et incomplètes : application en configuration experte, Raisonnement à partir de cas 1, Informatique et Systèmes d'Information », Hermes, Lavoisier, Paris, pp. 65-93, 2007.
 - [P&P 95] PU P. et PURVIS L., « Adaptation Using Constraint Satisfaction Techniques », Lecture Notes In Computer Science, vol. 1010, pp. 289-300, 1995.
 - [P&P 97] PU P. et PURVIS L., « Formalizing the adaptation process for Case-Based Design », Issues and Applications of Case-Based Reasoning in Design, Edited by Maher M. L. and Pu P., LEA Publishers, ch. 9, pp. 221-239, London, 1997.
 - [P&P 98] PU P. et PURVIS L., « COMPOSER: A Case Based Reasoning System for Engineering Design », Robotica, vol. 16, n° 3, pp. 285–295, 1998.
 - [P&S 04] PAL S. K. et SHIU S. C. K., « Foundations of Soft Case-Based Reasoning », Wiley-Interscience, pp. 136-160, New Jersey, 2004.
 - [P&T 95] PORTINALE L. et TORASSO P., « ADAPTER an Integrated Diagnostic System Combining Case-Based and Abductive Reasoning », Proceedings of 1st International Conference on Case-Based Reasoning, 1995.
 - [PAJ 01] PAJULA E., SEURANEN T. et HURME M., « Selection of separation sequences by casebased reasoning », Computer-Aided Chemical Engineering, vol. 9, pp. 469-474, 2001.

- [PAT 02] PATTERSON D., ROONEY N. et GALUSHKA M., « Efficient similarity determination and case construction techniques for case based reasoning », Proceedings of the 4th ECCBR, pp. 292-305, 2002.
- [PAT 04] PATTERSON D., ROONEY N. et GALUSHKA M., « An effective indexing and retrieval approach for temporal cases », Proceedings of the 17th International FLAIRS, AAAI press, 2004.
- [PIG 05] PIGNANI J. M., « Sistemas expertos », 2005, dans http://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_ano/orientadora1/mograias/pignani-sistemasexpertos.pdf
- [POL 66] POLANYI M., « The Tacit Dimension », Routledge & Kegan Paul Ltd, London, 1966.
- [PRO 93] PROSSER P., « An empirical study of phase transitions in binary constraint satisfaction problems », Artificial Intelligence, vol. 81, pp. 81-109, 1993.
- [PRO 11] PROSIM, software and services in process simulation, 2011, dans <http://www.prosim.net/>
- [PUR 98] PURVIS L., « Synergy and commonality in case based and constraint based reasoning », AAAI Spring Symposium on Multimodal Reasoning, 1998.
- [R&A 05] RICHTER M. et AAMODT A., « Case-Based reasoning foundations », The Knowledge Engineering Review., vol. 20, n° 3, pp. 203-207, 2005.
- [R&S 89] RIESBECK C. K. et SCHANK R. C., « Inside Case Based Reasoning », LEA, Inc., Hillsdale, New Jersey, 1989.
- [RED 96] REDDY S. Y., FERTIG K. W. et SMITH D. W. E., « Constraint management methodology for tradeoff studies ». Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, Irvine, California, 1996
- [REN 07A] RENAUD J., CHEBEL MORELLO B., FUCHS B. et LIEBER J., « Raisonement à Partir de Cas 1 : conception et configuration de produits », Ed. Lavoisier, Hermes, Informatique et systèmes d'information (IC2), 2007.
- [REN 07B] RENAUD J., CHEBEL MORELLO B., FUCHS B. et LIEBER J., « Raisonement à Partir de Cas 2 : surveillance, diagnostic et maintenance », Ed. Lavoisier, Hermes, Informatique et systèmes d'information (IC2), 2007.
- [REU 06] REUSSNER J C., « Optimisation de la livraison, du stockage, du mélange, et du chargement des pétroles bruts d'une raffinerie », Rapport de projet de fin d'études, INRIA, 2006.
- [RIO 00] RIOS I. D, RIOS I. S. et MARTIN J., « Simulation. Modelling and Applications », Alfaomega, Mexique, 2000.

-
- [ROD 99] RODA I.R., POCH M., SANCHEZ-MARRE M., CORTES U. et LAFUENTE J., « Consider case-based systems for control of complex processes », *Chemical Engineering Progress*, vol. 95, pp. 39-45, 1999.
 - [ROL 07] ROLDAN E., « Evaluación de alternativas para optimizar las rutas de distribución de helado de la empresa Fricongelados Citlaltepelt aplicando la simulación », Thèse de master à l'Institut Tecnológico de Orizaba, Mexique, 2007.
 - [ROL 10] ROLDAN E., NEGNY S., LE LANN J. M. et CORTES G., « Constraint Satisfaction Problem for Case-Based Reasoning Adaptation: Application in Process Design », *Computer-Aided Chemical Engineering*, vol. 28, pp. 397-402, 2010.
 - [ROL 11] ROLDAN E., NEGNY S., LE LANN J. M. et CORTES G., « Modified Case Based Reasoning cycle for Expert Knowledge Acquisition during Process design », *Computer-Aided Chemical Engineering*, vol. 29, pp. 296-300, 2011.
 - [RON 04] RONG B. G., KOLEHMAINEN E., TURUNEN I. et HURME M., « Phenomena-based methodology for process intensification », *Computer-Aided Chemical Engineering*, vol. 18, pp. 481-486, 2004.
 - [ROS 06] ROSSI F., VAN BEEK P. et WALSH T., « Handbook of Constraint Programming », Elsevier, 2006.
 - [RUE 02] RUET M., « Capitalisation et réutilisation d'expériences dans un contexte multi-acteur » Thèse de doctorat à l'Institut National Polytechnique de Toulouse, France, 2002.
 - [S&B 96] SURMA J. et BRAUNSCHWEIG B., « Case-base retrieval in process engineering: Supporting design by reusing flowsheets », *Engineering Applications of Artificial Intelligence*, vol. 9, pp. 385-391, 1996.
 - [S&B 00] STAHL A. et BERGMANN R., « Applying recursive CBR for the customization of structure products in an electronic shop », *Advances in Case Based Reasoning, 5th EWCBR, Lecture Notes in Artificial Intelligence*, pp. 1-15, 2000.
 - [S&B 03] SALIDO M.A. et BARBER F., « A constraint ordering heuristic for scheduling problems », *Proceeding of the 1st Multidisciplinary International Conference on Scheduling : Theory and Applications*, pp. 476-490, 2003.
 - [S&C 93] SMYTH B. et CUNNINGHAM P., « Complexity of Adaptation in Real-World Case-Based Reasoning Systems », *Proceedings of the 6th Irish Conference on Artificial Intelligence & Cognitive Science*, 1993.
 - [S&F 98] SQALLI M.H. et FREUDER E.C., « Integration of CSP and CBR to compensate for incompleteness and incorrectness of models », *AAAI-98 Spring Symposium on Multimodal Reasoning, March 23–25, Stanford University, California, USA*, 1998.

- [S&G 99] SOININEN T. et GELLE E. M, «Dynamic Constraint Satisfaction in Configuration », AAI Technical Report WS-99-05, 1999.
- [S&K 96] SMYTH B. et KEANE M. T., « Using adaptation knowledge to retrieve and adapt design cases », Knowledge-Based Systems, vol. 9, n° 2, pp. 127–135, 1996.
- [S&K 98] SMYTH B. et KEANE M. T., « CBR and the Similarity Assumption», Journal of Artificial Intelligence, vol. 102, n° 2, pp. 249-293, 1998.
- [S&M 99] DE SILVA GARZA A. G. et MAHER M. L., « An evolutionary approach to case adaptation ». Proceedings of 3rd International Conference on Case-Based Reasoning ICCBR-99, pp. 162–172, Monastery Seeon, Munich, Germany 1999.
- [S&W 86] STANFILL C. et WALTZ D., « Towards memory-based reasoning », Communications of the Association for Computing Machinery, ch. 29, pp. 1213-1228, 1986.
- [SAM 95] SAM J., « Constraint consistency techniques for continuous domains », Thèse de doctorat à l’Ecole Polytechnique Fédérale de Lausanne, Suisse, 1995.
- [SCA 04] SCARAVETTI D., « Formalisation préalable d’un problème de conception pour l’aide à la décision en conception préliminaire », Thèse de doctorat à l’Ecole Nationale Supérieure d’Arts et Métiers, Bordeaux, 2004.
- [SCH 82] SCHANK R. C., « Dynamic memory: A theory of learning in people and computers », Cambridge: Cambridge University Press, 1982.
- [SCH 83] SCHÖN D. A., « The reflective practitioner », Sage, London, 1983.
- [SCH 97] SCHMITT G., DAVE B. et SHIH S., « Case-based architectural design : the experience of CADRE », Introduction to the Issues and Applications of Case-Based Reasoning in Design, LEA Publishers, ch. 10, pp. 241-260, London, 1997.
- [SCH 99] SCHANK R. C., « Dynamic Memory Revisited », Cambridge University Press, 1999.
- [SEI 04] SEIDER W. D., SEADER J. D. et LEWIN, D. R., « Product and Process Design Principles: Synthesis, Analysis and Evaluation », 2nd Edition, Wiley, 2004.
- [SEI 10] SEIDER W. D., SEADER J. D. et LEWIN, D. R., « Product and Process Design Principles: Synthesis, Analysis and Evaluation », 3rd Edition, Wiley, 2010.
- [SER 87] SERRANO D. « Constraint Management in Conceptual Design», Thèse de doctorat à Massachusetts Institute of Technology, USA 1987.
- [SER 06] SERRAFERO P., GOMES S., BONNINVAR D. et JEZEQUEL L., « De la mémoire projet à la compétence métier : vers la synthèse de connaissances métier en ingénierie robuste des produits/process », Conférence Internationale IDMME, France, 2006
- [SEU 06] SEURANEN T., « Studies on Computer-Aided Conceptual Process Design », Thèse de doctorat à l’Université de Technologie d’Helsinki, Finlande, 2006.

-
- [SFG 10] SFGP, « La génie des procédés », Société française de génie des procédés, dans <http://www.sfgp.asso.fr/>
 - [SHA 98] SHAPIRO R., FELDMAN Y. A. et DECHTER R., « On the Complexity of Interval-Based Constraint Networks », Rapport Technique, n° 76, University of California, Irvine, 1998.
 - [SIM 92] SIMOUDIS E., « Using case-based retrieval for customer technical support », IEEE Expert, vol. 7, n° 5, pp. 7-13, 1992.
 - [SMI 05] SMITH R., « Chemical Process, Design and Integration », John Wiley & Sons, England, 2005.
 - [SMY 93] SMYTH B., « Complexity of Adaptation in Real-World Case-Based Reasoning Systems » Proceedings of the 6th Irish Conference on AI & Cognitive Science, pp. 229-240, 1993.
 - [SQA 99] SQALLI M.H., PURVIS L. et FREUDER E. C., « Survey of applications integrating constraint satisfaction and case-based reasoning », 1st International Conference and Exhibition on the Practical Application of Constraint Technologies and Logic Programming (PACLP99), April 19–21, London, UK, 1999.
 - [SUH 90] SUH N. P., « The principles of design », Oxford University Press, New York, 1990.
 - [SUL 02] O’SULLIVAN B., « Constraint-Aided Conceptual Design », Engineering Research Series 9, Professional Engineering Publishing Limited, UK, 2002.
 - [SYC 87] SYCARA E. P., « Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods », Thèse de doctorat de Georgia Institute of Technology, 1987.
 - [T&R 05] TONIDANDEL F. et RILLO M., « Case adaptation by segment replanning for case-based planning systems », Case-Based Reasoning Research and Development, pp. 579–594, 2005.
 - [TOM 02] TOMIYAMA T., YOSHIOKA M. et TSUNAYA A., « A knowledge operation model of synthesis », Engineering design synthesis, Springer, Londres, pp. 67-90, 2002.
 - [TRA 97] TRAVE-MASSUYES L., DAGUE P. et GUERRIN F., « Le raisonnement qualitatif pour les sciences de l’ingénieur », Éd. Hermès, 1997.
 - [U&E 00] ULRICH K. et EPPINGER S., « Product Design and Development », McGraw-Hill, New York, 2000.
 - [V&C 93] VELOSO M. M. et CARBONELL J. G. « Derivational Analogy in PRODIGY: Automating Case Acquisition, Storage, and Utilization », Machine Learning, vol. 10, pp. 249-278, 1993.

- [VAR 05] VAREILLES E., « Conception et approches par propagation de contraintes : contribution à la mise en œuvre d'un outil d'aide interactif », Thèse de doctorat à l'Institut National Polytechnique de Toulouse, France, 2005.
- [VAR 07] VAREILLES E., ALDANONDO M. et GABORIT P., « Evaluation and design: a knowledge-based approach », *International Journal of Computer Integrated Manufacturing*, vol. 20, n° 7, pp. 639–653, 2007.
- [VAR 11] VAREILLES E., ALDANONDO M., CODET DE BOISSE A. et COUDERT T., « How to take into account general and contextual knowledge for interactive aiding design: Towards the coupling of CSP and CBR approaches », *Engineering Applications of Artificial Intelligence*, vol. 25, pp. 31–47, 2011.
- [VEL 94] VELOSO M. M., « Planning and learning by analogical reasoning », *Lecture Notes in Computer Science*, n° 886, Springer, Berlin, 1994.
- [VER 04] VERNAT Y., « Formalisation et qualification de modèles par contraintes en conception préliminaire », Thèse de doctorat à l'École Nationale Supérieure d'Arts et Métiers, Centre de Bordeaux, France, 2004.
- [W&B 98] WILKE W. et BERGMANN R., « Techniques and Knowledge Used for Adaptation During Case-Based Problem Solving », *Tasks and Methods in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence*, vol. 1416, pp. 497-505, 1998.
- [W&F 92] WALLACE R. J. et FREUDER E. C., « Ordering heuristics for arc consistency algorithms », In *AI/GI/VI '92*, pp. 163–169, 1992.
- [W&N 01] WIBOWO C. et NG K.M., « Product oriented process synthesis and development : creams and pastes », *American Institute of Chemical Engineering Journal*, vol. 47, pp. 2746-2767, 2001.
- [WAL 07] WALLACE M., « Constraint Programming – the paradigm to watch », *Constraint Programming Letters*, vol. 1, pp. 7-13, 2007.
- [WAN 09] WANG X., ZHENG J. et ZENG Q., « A Design of Product Collaborative Online Configuration Model », Y. Luo (Ed.): *CDVE 2009, Lecture Notes in Computer Science*, vol. 5738, pp. 359–366, Springer Berlin, 2009.
- [WAT 97] WATSON I., « Applying Case-Based Reasoning: Techniques for Enterprise Systems », Morgan Kaufmann Publishers, 1997.
- [WAT 99] WATSON I., « Case-based reasoning is a methodology, not a technology », *Knowledge-Based Systems*, vol. 12, pp. 303–308, 1999.
- [WIL 92] WILLIAMS B., « Interaction-based design: constructing novel devices from first principles », *Intelligent Computer Aided Design*, Elsevier, North Holland, pp. 255-274, 1992.

-
- [WIL 97] WILKE W., VOLLRATH I., ALTHOFF K. D. et BERGMANN R., « A framework for learning adaptation knowledge based on knowledge light approaches », Proceedings of the 5th German Workshop on Case-Based Reasoning, pp. 235–242, 1997.
 - [WIL 98] WILKE W., SMITH B. et CUNNINGHAM P., « Using configuration techniques for adaptation », Case-Based Reasoning Technology from Foundations to Applications (M. Lenz, B. Bartsch-Spörl, H. D. Burkhard, and S.Wess, eds.), Springer-Verlag, Berlin, pp. 139–168, 1998.
 - [WIT 09] WHITE J., DOUGHERTY B., SCHMIDT D.C. et BENAVIDES D., « Automated reasoning for multi-step software product-line configuration problems », Proceedings of the Software Product Line Conference, pp. 11–20, 2009.
 - [XIA 97] XIA Q., RAO M., HENRICKSSON C. et FARZADEH H., « Case-based reasoning for intelligent fault diagnosis and decision making in pulp processes », Pulp & Paper Canada, vol. 98, pp. 26-30, 1997.
 - [XUE 06] XUEREB C., POUX M. et BERTRAND J., « Agitation et mélange : Aspects fondamentaux et applications industrielles », Techniques et Ingénierie, Dunoud/L'usine Nouvelle, 2006.
 - [YAN 01] YANNOU B., « Préconception de produits », Habilitation à Diriger des Recherches, Institut National Polytechnique de Grenoble, 2001.
 - [Z&D 06] ZHA X. F. et DU H., « Knowledge-intensive collaborative design modelling and support Part I : Review, distributed models and framework », Computers in Industry, vol. 57, pp 39-55, 2006.
 - [Z&R 05] ZIV-AV A. et REICH Y., « SOS-subjective objective system for generating optimal product concepts », Design Studies, vol. 26, Springer, pp.509-533, 2005.
 - [Z&R 10] ZAVBI R. et RIHTARSIC J., « Synthesis of elementary product concept based on knowledge twisting », Research in Engineering Design, vol. 21, Springer, pp. 69-85, 2010.
 - [Z&G 01] ZACKLAD M. et GRUNDSTEIN M., « Ingénierie et Capitalisation des connaissances », Traité IC2, Paris, Hermès, 2001.
 - [ZAG 11] ZAGHLOUL O., « La programmation par contraintes », 2011, dans <http://www.inventive-design.net/content/view/295/1/>
 - [ZHO 10] ZHOU M., CHEN Z., HE W. et CHEN X., « Representing and matching simulation cases : A case-based reasoning approach », Computers & Industrial Engineering, vol. 59, pp. 115-125, 2010.

Annexe 1

Algorithme d'indexation sphérique

Cette annexe présente la description de l'algorithme d'indexation sphérique basé sur une discrétisation de la base de cas.

Annexe 1 : Algorithme d'indexation sphérique

Notation

B : base de cas

d : distance

QC : cas cible

SC^j : cas source j dans B

Indices : **c** : distance entre deux cellules,

i : attribut du cas considéré (QC ou SC^j),

s : distance de recherche autour de QC,

t : distance réelle entre QC et une cellule

Algorithme d'indexation sphérique

Trouver le sous ensemble final de cas pertinents (FS), revient à déterminer les cas au voisinage de QC. Pour chaque dimension i , d_i délimite un intervalle de valeurs acceptables autour de l'attribut QC_i : $[QC_i - d_i ; QC_i + d_i]$. Seuls les cas avec $SC_i \in [QC_i - d_i ; QC_i + d_i]$ sont à conserver. L'intersection de ces intervalles sur chacune des dimensions définit FS : $FS = \{ SC^j \in B / \forall i SC_i \in [QC_i - d_i ; QC_i + d_i] \}$

Etape 1 : $Cellsubset_1 = \{ Cellule_i / Etiquette \text{ de la cellule } i \leq d_s^2 \}$

Dans une première étape le raisonnement se focalise sur les cellules créées suite à la discrétisation et non sur les cas contenus à l'intérieur. La distance relative minimum entre la cellule centre (CC : cellule contenant QC) et les autres cellules de B est calculée, elle correspond à la distance minimum d_c entre deux cas pris dans chacune d'elles. Pour une cellule arbitraire, cette distance peut être précalculée et stockée. La figure A1.1a présente un exemple en 2 dimensions. Une fois QC définit, CC est connue et le modèle précédent est appliqué en le centrant sur CC. Sur ce modèle les cellules sont étiquetées avec un nombre correspondant à d_c^2 . Seules les cellules avec une étiquette inférieure ou égale à d_s^2 sont conservées. La figure A1.1b illustre ce propos pour $d_s^2 = 4,1$, toutes les cellules avec une étiquette supérieure ou égale à 4 sont automatiquement exclues (cellules retenues sont à l'intérieur du périmètre rouge).

Etape 2 : Test $Dec(d_s) < \xi_k$. Utilisation du vecteur adéquate pour les cellules avec étiquette $\geq Ent(d_s^2)$. $Cellsubset2 = Cellsubset1 - \text{cellules rejetées}$.

Jusqu'à présent seule la cellule contenant QC a été considérée, mais certaines cellules ont une distance inférieure à d_s par rapport à QC et non pas à sa cellule. Sur l'exemple précédent les cellules étiquetées d'un 4 en gras (à l'extérieur du cercle grisé) sont à rejeter. Par conséquent la position de QC dans sa cellule est intégrée dans l'algorithme grâce à sa distance ξ_k par rapport à chacune des frontières de sa cellule, figure A.1.2 en 2 dimensions. Suite à une démonstration mathématique, on obtient l'ensemble des cellules à enlever grâce au test suivant :

$Dec(d_s) < \xi_k$ ou non Test effectué que lorsque QC connu.

Ce test doit prendre en considération toutes les positions possibles de QC dans CC, soit 2^{Na} combinaisons conduisant à autant de vecteurs (précalculés) contenant les cellules à conserver. Toutefois, toutes les cellules de $Cellsubset1$ ne sont pas à tester, certaines satisfont automatiquement le test. Dans la réalité les étapes 1 et 2 sont exécutées simultanément, la distance d_c augmente progressivement et lorsqu'elle atteint $Ent(d_s)$, le test est activé avec la sélection du vecteur adéquate.

Etape 3 : Pour chaque cellule de $Cellsubset2$ dont $d_t > d_s - 1$. Test : $d_t > d_s$?. $Cellsubset3 = Cellsubset2 - \{Cell_i / d_t > d_s\}$

A cause d'une hypothèse dans la démonstration mathématique, le test précédent ne remplit pas strictement son rôle, car certaines cellules satisfaisant le test ont en réalité une distance supérieure à d_s par rapport à la localisation de QC. Dans l'exemple précédent c'est le cas de la cellule identifié par ② sur la figure A1.1b. Dans ces conditions, ces cellules sont à exclure au prix d'un test supplémentaire intégrant la distance réelle d_t . Dans cette étape également toutes les cellules ne sont pas à tester seulement celles dont la distance est supérieure à $d_s - 1$ (heuristique).

Etape 4 : $FCS = Cellsubset3 - \text{Cellules vides}$. $FS = \{SC^j / SC^i \in Cell_i \text{ avec } Cell_i \in FCS\}$

Seuls les cas contenus dans l'ensemble final des cellules sont soumis au calcul de la similarité et/ou de l'adaptabilité. A noter que, comme les cas sont inégalement répartis, l'algorithme intègre l'éventualité d'avoir des cellules vides et donc de les éliminer automatiquement.

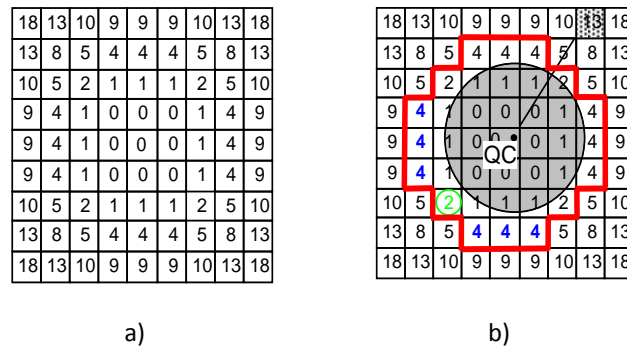


Figure A1.1 Modèle pour distance minimum entre cellules

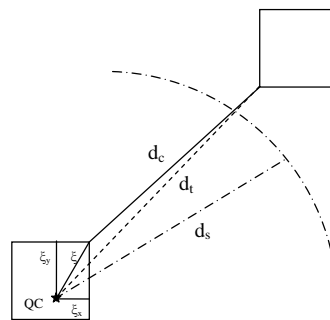


Figure A1.2 Position de QC dans sa Cellule

Annexe 2

Calcul de la similarité locale

Cette annexe donne un aperçu sur le calcul de la similarité locale.

Annexe 2 : Calcul de la similarité locale

Pour les attributs numériques, le calcul de la distance se base sur la formule suivante :

$$d(x_i, y_i) = |x_i - y_i|$$

Or cette façon de mesurer risque de fausser le résultat lorsque les attributs ont des domaines de valeurs de tailles différentes (ce qui est le cas en génie des procédés avec des attributs du type pression et température). Il est donc nécessaire de normaliser le calcul de la distance. Une solution consiste alors à exprimer explicitement le domaine de définition et d'intégrer cette expression dans le calcul. Ceci est possible avec la fonction Int_i qui calcule la différence entre les valeurs maximum et minimum de l'attribut :

$$\text{sim}_{\text{loc}}(x_i, y_i) = 1 - \frac{|x_i - y_i|}{\text{Int}_i}$$

Pour ce qui est de la similarité locale pour des attributs textuels (ou des ensembles textuels comme par exemple la composition d'un mélange), la similarité locale est souvent calculée de façon binaire :

$$\text{sim}_{\text{loc}}(x_i, y_i) = \begin{cases} 1 & \text{si } x_i = y_i \\ 0 & \text{si } x_i \neq y_i \end{cases}$$

Toutefois, cette méthode de calcul de similarité est perfectible dans le cas de mélange de composés chimiques par exemple. Dans la méthode retenue, le mélange de constituants est représenté par un seul attribut. Bien évidemment, chaque cas à un mélange spécifique composé d'un nombre et de constituants différents. La similarité locale entre deux attributs « mélange » se calcule alors en deux étapes :

1. Construction d'une matrice de similarité locale binaire.

Pour chacun des binaires composé d'un constituant du problème cible (i) et d'un constituant du problème source (j), une similarité locale binaire est calculée ($\text{bsim}_{i,j}$). Le calcul de cette similarité locale binaire s'appuie sur une structure arborescente. Elle décompose l'ensemble des composés chimiques sous forme d'arbre avec des ramifications successives et affecte une valeur de la similarité à chaque nœud de l'arbre (entre 0 et 1). La similarité locale binaire est affectée en fonction du dernier nœud de ramification commun entre les deux composés (plus le nœud est bas dans la ramification, plus la similarité est forte), figure A2.1. Pour chaque binaire possible, la similarité locale binaire est estimée et stockée dans une matrice, illustrée sur le tableau A2.1.

2. Calcul de la similarité locale

Durant cette étape on cherche à sélectionner les binaires les plus similaires grâce à la

résolution du système suivant : $\text{sim}_{\text{loc}}(x_i, y_i) = \text{Max}_{\alpha_{ij}} \left(\frac{1}{m} \sum_{i=1}^{nt} \sum_{j=1}^{ns} \alpha_{ij} \text{bsim}_{ij} \right)$ avec $\forall j \in [1, ns] \sum_{i=1}^{nt} \alpha_{ij} \leq 1$

et $\forall i \in [1, nt] \sum_{j=1}^{ns} \alpha_{ij} \leq 1$

α_{ij} binaire valant 1 si le binaire (i,j) est retenu, 0 sinon (équivalent d'une variable d'affectation), $m = \max(nt, ns)$ car le nombre de constituants du mélange cible (nt) peut être différent de celui du mélange source (ns). Les deux contraintes matérialisent le fait qu'un composé (cible ou source) ne peut être affecté que dans un seul binaire. Pour l'exemple du tableau A2.1, $\text{sim}_{\text{loc}} = 0.525$ et les similarités locales binaires retenues sont encerclées (sur cet exemple plusieurs affectations sont possibles).

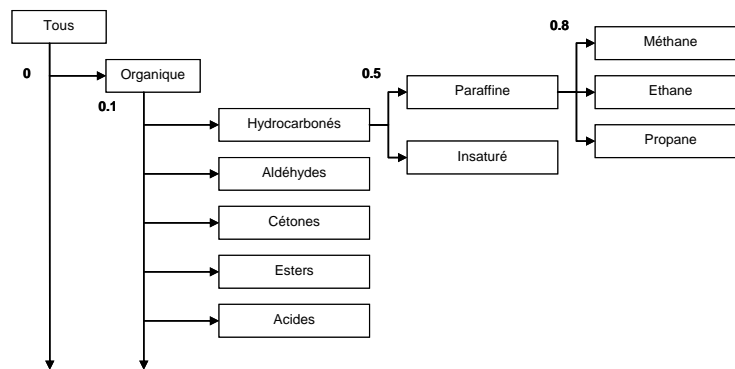


Figure A2.1 Structure Arborescente

Source \ Cible	Ethanol	Eau	Acide Acétique	Acetate d'Ethyl
Méthanol	$\text{bsim}_{ij}=0.9$	0.1	0.1	0.1
Ethanol	1	0.1	0.1	0.1
Eau	0.1	1	0.1	0.1

Tableau A2.1 Exemple de calcul de similarité locale

A noter que dans l'outil, les valeurs affectées à chaque nœud de l'arbre présenté sur la figure sont paramétrables. D'autres fonctions de similarités (non présentées ici) sont définissables et permettent de personnaliser le RàPC et ainsi de refléter la manière de voir la similarité locale entre deux attributs (des fonctions *ad hoc* peuvent être définies ponctuellement).

Utilisation des sous ensembles flous pour matérialiser l'imprécision

Dans cette annexe on montre l'utilisation des ensembles flous pour matérialiser l'imprécision sur les valeurs des attributs dans la représentation des cas.

Annexe 3 : Utilisation des sous ensembles flous pour matérialiser l'imprécision

Définition des sous ensembles

Les ensembles flous permettent de définir des degrés d'appartenance d'un élément à un ensemble. Un ensemble flou sur un domaine est défini à l'aide d'une fonction caractéristique μ_s représentant le degré d'appartenance d'un élément z à l'ensemble. Les ensembles flous sont représentés à l'aide de fonctions d'appartenance triangulaires (définies à l'aide de trois paramètres : d_i , c , d_s) ou trapézoïdale (définies par 4 paramètres : d_i , c_1 , c_2 , d_s), figures A3.1.

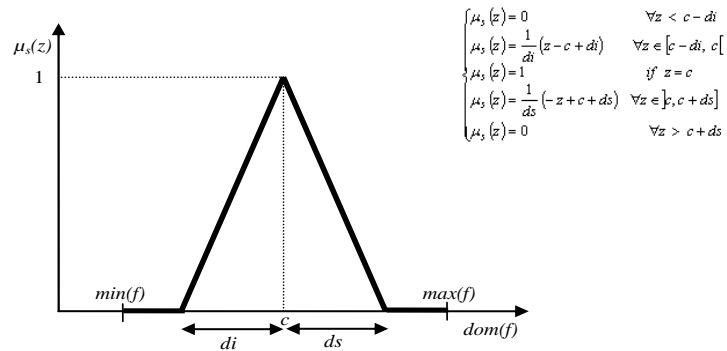


Figure A3.1a Distribution triangulaire

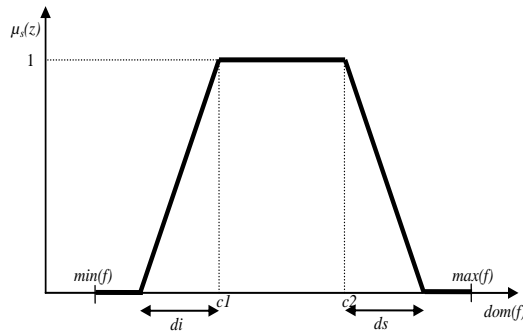


Figure A3.1b Distribution trapézoïdale

Construction des sous ensembles

Pour décrire l'imprécision relative à un problème, il faut valuer l'attribut considéré (attribut= v), spécifier un paramètre λ traduisant l'imprécision (pourcentage) autour de la valeur fixée et enfin stipuler une relation. Cette dernière sert à orienter la recherche par

rapport à la valeur spécifiée, six relations sont possibles : *inférieur*, *inférieur ou égal*, *égal*, *supérieur*, *supérieur ou égal* et *entre* (lors de l'utilisation de cette dernière relation, il faut spécifier 2 valeurs pour valuer l'attribut). A titre d'exemple, la relation *égal* exprime le fait que z devra être environ égale à v mais dans l'intervalle $[v - \lambda v ; v + \lambda v]$. La représentation trapézoïdale sert à modéliser la relation *entre* tandis que la triangulaire sert pour toutes les autres relations. Grâce aux informations précédentes la construction des sous ensembles flous est alors possible à l'aide du calcul des paramètres (d_i , d_s , c ou c_1 et c_2) comme indiqué dans les tableaux A3.1 et illustré sur la figure A3.2 (pour *égale*).

	Egal	Sup, Sup-égal	Inf, Inf-égal
C=	v	$Max(f)$	$min(f)$
Di=	$Min(\lambda v ; v - min(f))$	$Min(max(f) - (v - \lambda v) ; max(f) - min(f))$	0
Ds=	$Min(\lambda v ; max(f) - v)$	0	$Min((v + \lambda v) - min(f) ; max(f) - min(f))$

Tableau A3.1a Paramètres pour la fonction caractéristique pour la représentation triangulaire

Entre	
c1=	$v1$
c2=	$v2$
Di=	$min(\lambda v1 ; v1 - min(f))$
Ds=	$min(\lambda v2 ; max(f) - v2)$

Tableau A3.1b Paramètres pour la fonction caractéristique pour la représentation trapézoïdale

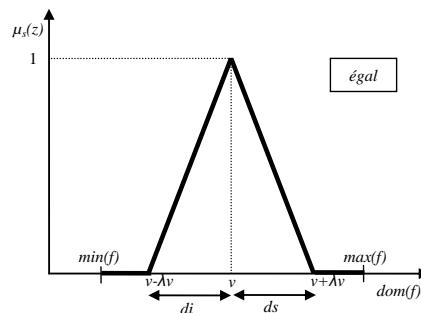


Figure A3.2 Forme de fonction d'appartenance (relation égale)

Calcul de la similarité

Après construction des sous ensembles flous pour chacune des valeurs imprécises, la similarité locale peut être calculée : $sim_{loc}(v,z) = \mu_s(z)$ où v est la valeur de l'attribut du problème cible et z celle du même attribut dans le cas source.

Le calcul de la similarité globale est alors personnalisable afin d'exploiter au mieux les connaissances passées. Outre les fonctions de similarité locales, la vision de l'utilisateur peut être prise en compte par la pondération de chaque attribut. L'affectation des poids peut être soit directement décidée par l'utilisateur en spécifiant une valeur, soit réalisée par classement des attributs par ordre d'importance selon un rang (aide à l'attribution des poids). Les poids de chacun des attributs sont évalués par (rang 1 pour l'attribut le plus important) :

$$w_i = 1 - \frac{\text{rang}_i - 1}{\text{Max}(\text{rang}_i)} \quad (w_i \text{ étant compris entre } [0;1])$$

Calcul de l'adaptabilité d'un cas

Cette annexe propose une méthode générique d'évaluation d'un critère d'adaptabilité dont le principe est de lier la facilité d'adaptation d'un cas au potentiel de valeurs solutions que contient son espace d'adaptation.

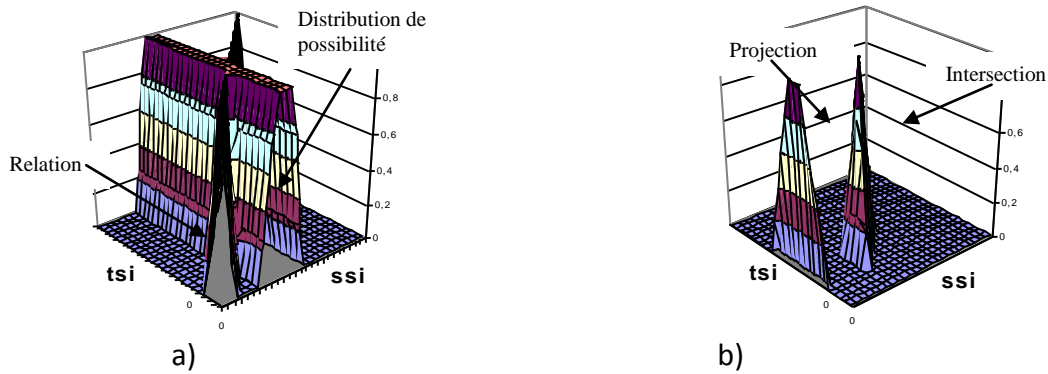
Annexe 4 : Calcul de l'adaptabilité d'un cas

Construction de l'espace d'adaptation

L'espace d'adaptation correspond à l'espace dans lequel la solution du problème cible devra être recherchée. Il se construit à partir des attributs de la solution du cas source mémorisé et des connaissances supplémentaires fournies par l'utilisateur. L'espace d'adaptation comprend l'ensemble des domaines d'adaptation de chacun des attributs solution. Lors de l'étape de mémorisation d'un nouveau cas, les attributs de la partie solution ne sont pas obligatoirement stockés avec une valeur unique mais l'expert peut en compléter la description en spécifiant l'ensemble des valeurs possible. En effet, il est fréquent, surtout en conception préliminaire, que certains attributs ne soient pas fixés à une valeur précise mais que plusieurs valeurs soient admissibles. Dans ces conditions ils sont renseignés à l'aide d'une distribution de possibilité, formalisée par un ensemble flou de forme trapézoïdale.

Au niveau de l'adaptation, il est aussi important de prendre en compte les expériences passées et les connaissances de l'utilisateur afin d'apporter une réponse pertinente à son problème. Ainsi, il doit pouvoir exprimer son point de vue et ses préférences sur les valeurs d'une solution. Grâce à des relations analogues aux fonctions de similarité locale, l'utilisateur précise comment exploiter chaque attribut de la solution. A titre d'exemple, pour un attribut i de la solution source, si l'utilisateur estime que dans sa solution l'attribut devra comporter une valeur « proche » de celle-ci, il peut modéliser son opinion grâce à la relation égale et un pourcentage λ_s (explicitation d'une « similarité » entre les attributs de la partie solution des cas). Pour chaque attribut solution i , la relation μ_{soli} permet de situer la valeur recherchée pour la solution cible (ts_i) par rapport à la solution source (ss_i). La figure A4.1a illustre cet aspect pour la relation égale ; ts_i doit être « proche » de ss_i (égale à $\pm\lambda_s$ pourcent).

Pour construire le domaine d'exploitation d'un attribut, il faut tirer profit de la distribution de possibilité de l'attribut de la solution source et de la fonction de similarité (relation). La projection sur ts_i de l'intersection entre la fonction de similarité μ_{soli} et la distribution de possibilité de ss_i permet d'obtenir la distribution de possibilité de ts_i (ensemble des valeurs possibles similaires à ss_i). La figure A4.1b présente une illustration graphique de cette intersection et de cette projection.



Figures A4.1 Représentation graphique de la construction des domaines d'adaptation

Après projection, l'ensemble obtenu regroupe les valeurs similaires à ss_i selon la fonction de similarité propre à cet attribut. Pour affiner le domaine d'adaptation seules les valeurs les plus similaires seront conservées. En conséquence une α coupe de cet ensemble est calculée (α à fixer par l'utilisateur), définissant ainsi l'ensemble des valeurs du domaine d'adaptation.

Calcul de l'adaptabilité

La taille du domaine d'adaptation traduit la facilité d'adaptation de cet attribut. Plus ce domaine est large plus l'attribut du cas source sera facile à adapter, car la probabilité de trouver une valeur acceptable est plus élevée. Il faut donc tirer profit de cette taille pour mesurer l'adaptabilité. La mesure de spécificité d'un ensemble flou permet de traduire le degré de flou de cet ensemble. Plus l'ensemble est spécifique moins il contient de valeur plus sa spécificité tend vers 1 ($Sp=1$ pour le cas d'un ensemble avec une valeur unique). A l'inverse la spécificité tend vers 0 pour les ensembles comprenant de nombreuses valeurs.

$$S_p(D_i) = \int_0^1 \frac{1}{\sup_{\alpha} D_i - \inf_{\alpha} D_i} d\alpha$$

$\sup_{\alpha} D_i$ (resp. $\inf_{\alpha} D_i$) représente la limite supérieure (resp. inférieure) de la α coupe sur le domaine D_i

L'adaptabilité de l'attribut peut ainsi être évaluée par : $ad_i = 1 - Sp(D_i)$

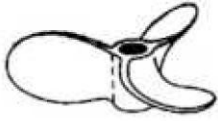
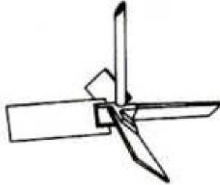

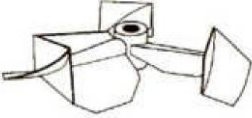
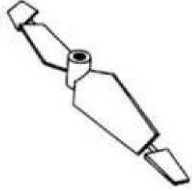
Et l'adaptabilité du cas source par : $ad_s = \sum_{i=1}^n ad_i / n$

Les différents types de mobiles d'agitation

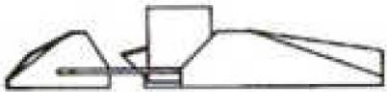

Une liste avec les différents types de mobiles d'agitation est présentée dans cette annexe.

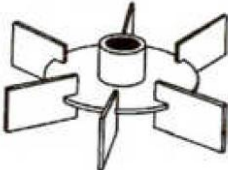
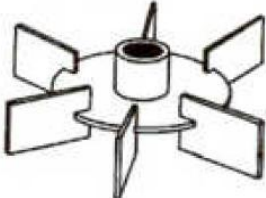



Annexe 5 : Les différents types de mobiles d'agitation

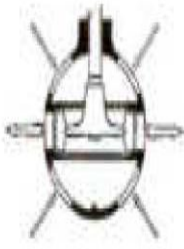
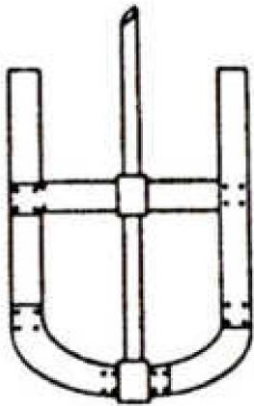
Remarque : Le nombre de pales n_p sera déterminé lors de la conception détaillée.

Mobiles d'agitation à débit axial	
<i>Hélice marine à 3 ou 4 pales</i>	
<p><i>Turbine à n_p pales inclinées</i></p> <p>$n_p = 3$ ou 4 type <i>TA3, TA4</i></p> <p>$n_p = 3$ type <i>http</i></p> <p>$n_p = 4$ type <i>HAP</i></p> <p>$n_p = 4$ type <i>A 200</i></p> <p>$4 \leq n_p \leq 6$ type <i>Ekato</i> et type <i>Robin</i></p> <p>$n_p = 2, 4$ inclinaison 45°</p> <p>ou 6</p> <p>$n_p = 4$ type quadripale inclinaison 40°</p> <p>$n_p = 6$ type <i>Pfaudler 30^\circ</i></p> <p>$n_p = 3$ ou 4 type <i>TPI</i></p> <p>$2 \leq n_p \leq 6$ type <i>TPI</i></p>	 
<p><i>Hélice à double flux à n_p pales</i></p> <p>$n_p = 2$ type <i>HPM</i> et <i>DF</i></p> <p>$n_p = 2$ type <i>DF 2</i></p> <p>$n_p = 2$ type <i>HFI</i></p> <p>$n_p = 2$ Organes <i>MIG</i></p> <p>$n_p = 2$ Organes <i>INTERMIG</i></p> <p>$n_p = 2$ <i>Sabre DF</i></p> <p>$n_p = 3$ type <i>LB, LC</i> et <i>DF</i></p> <p>$n_p = 3$ type biflux centripète</p> <p>$n_p = 3$ type <i>DF3</i></p>	 

<p><i>Hélice à n_p pales à profil mince</i></p> <p>$n_p = 2$ type <i>Interprop, Isojet</i></p> <p>$n_p = 2$ ou 3 type <i>GPM</i> à pales minces ou courbes</p> <p>$n_p = 3$ type <i>HAS</i></p> <p>$n_p = 3$ Hélico-mélangeur type <i>Sabre (8T, 31T, S, R, C)</i></p> <p>$n_p = 3$ multiplan tripale types (<i>TT, TTA, TTP, TTPA, TTM</i>)</p> <p>$n_p = 3$ type <i>LA, LB, LC</i></p> <p>$n_p = 3$ type <i>A 310</i></p> <p>$n_p = 2$ ou 3 type bipale ou tripale profilée</p> <p>$n_p = 3$ à doigts de turbulence</p> <p>$n_p = 3$ type <i>GPP, GPE, GPL</i></p> <p>$n_p = 3$ type <i>A 6000</i> en matériau composite</p> <p>$n_p = 3$ type <i>HPM 10 et 20</i></p> <p>$n_p = 3$ type <i>HTPG4</i></p> <p>$n_p = 4$ type <i>HPM 10, 20, 30, 40, 50 et 60</i></p> <p>$n_p = 4$ type <i>A 315</i> à 4 pales larges</p> <p>$n_p = 3$ Hélice à 3 pales à 30°</p> <p>$n_p = 6$ Hélice de compression</p> <p>$n_p = 2, 3$ ou 4 type <i>Viscoprop</i></p>	
<p><i>Mobile pour fluide visqueux</i></p> <p>Turbine hélicoïdale type <i>A 400</i></p> <p>Organe <i>Ekato-Hélicoïdal</i> en 1 ou 2 spirales avec ou sans vis intérieure</p> <p>Ruban simple ou double avec ou sans vis intérieure</p> <p>Vis d'Archimède à double ou simple spirale</p> <p>Hélice type <i>A 320</i> à 3 pales larges</p> <p>Ruban simple ou double avec ou sans vis d'Archimède</p> <p>Organe <i>Ekato-Paravisc</i></p> <p>Vis d'Archimède planétaire</p>	

<p><i>Mobile à disque et à 3 pales inclinées à 45° à leur extrémité refoulant le liquide de bas en haut, utilisé pour l'agitation de fond de cuve type TFC ou DEL type DFC ou EFC</i></p>	
<p><i>Agitateur submersible multidirectionnel Hélice à 2 pales série pale banane ou à 3 pales série SR 4600 Mobile Aquajet à 2 pales</i></p>	

Mobiles d'agitation à débits radial et tangentiel	
<p><i>Turbine à disque type Rushton à n_p pales droites, inclinées, incurvées, ou concaves $4 \leq n_p \leq 12$</i></p>	
<p><i>Turbine à pales droites ou incurvées $2 \leq n_p \leq 6$</i></p>	
<p><i>Turbine fermée aspirant axialement et refoulant Radialement type Bicône Turbine centripète aspirant axialement et refoulant radialement dans deux directions opposées</i></p>	
<p><i>Turbine fermée à disque et à pales courbes avec ou sans contre-pièces fixes type TA Rotor centrifuge ou centripète avec stator alvéolé ou à Grille Rotor flasque à aubes courbes type ME Turbine auto-aspirante semi-fermée</i></p>	
<p><i>Turbine à pales courbes avec un système de contre-pièces fixes</i></p>	

<p><i>Mobile de dispersion</i> type SC type disque <i>Ekato Mizer</i> type défloculeuse type turbine de cisaillement types R 500 et R 510 à barres Turbine à dents pour cisaillement élevé type disperseuse et type sevin à ouïes</p>	
<p><i>Agitateur à ancre (ou cadre), à flux tangentiel pour milieux visqueux et raclage de paroi</i> type R 400 type arbre <i>Ekato</i> type MA, AFI type ruban (simple ou double) et vis d'Archimède type pale ancre à racleurs démontables et orientables type <i>Multimix</i> (à 2 mouvements coaxiaux) type <i>Trimix</i> (à 3 mouvements coaxiaux) type cadre raclant type ancre (avec ou sans racleurs) type ME</p>	
<p><i>Agitateur à barrière, à flux tangentiel pour agitation douce</i> type herse Hélices haubanées type MAB</p>	