

Université Badji Mokhtar d'Annaba  
Département d'Informatique  
Master 1 IATI 2019/2020  
Méthodes de l'intelligence artificielle  
Chargé Module : **Dr Djebbar Akila**

## Chapitre 2

# Le Raisonnement à Partir de Cas (RàPC)

### Plan du cours

1. Introduction
2. Approche du Raisonnement à Partir de Cas
  - 2.1. Historique
  - 2.2. Les origines des systèmes RàPC
  - 2.3. Représentation d'un cas
3. Composantes d'un système de Raisonnement à Partir de Cas
  - 3.1. Processus
    - 3.1.1. Remémoration
    - 3.1.2. Adaptation ou réutilisation
    - 3.1.3. Maintenance ou révision
    - 3.1.4. Construction ou apprentissage
  - 3.2. Connaissances dans un système à base de cas
4. Organisation de la mémoire
  - 4.1. Organisation plate
  - 4.2. Réseaux à caractéristiques partagées
  - 4.3. Réseaux de discrimination
  - 4.4. Réseaux redondants de discrimination
  - 4.5. Exemple de modèles hybrides de mémoire de cas
5. Exemples des systèmes RàPC
6. Conclusion

## 1. Introduction

Le raisonnement en Intelligence Artificielle a souvent été synonyme d'inférence par règles ou par modèles. Et ce malgré l'existence d'une grande diversité de modes de raisonnement à l'image de la richesse des mécanismes mentaux de l'homme. La majeure partie des applications en Intelligence Artificielle consiste à reproduire le raisonnement humain. Le raisonnement par cas est une approche de résolution de problème basée sur l'utilisation d'expériences passées appelées cas.

Le Raisonnement à Partir de Cas « RàPC » (ou *Case Based Reasoning* : C.B.R.) gère une mémoire pour stocker ces différents cas. Pour résoudre un nouveau problème, il commence par rechercher dans cette mémoire le cas le plus proche de ce problème. Par la suite le système adapte l'ancien cas au problème nouveau pour en déduire une solution à partir de la solution de l'ancien cas [AAM 94].

D'un point de vue cognitif, le raisonnement par cas est à la base de théories psychologiques du comportement humain, en particulier lors de la prise de décision. Or, le recours à des situations passées pour résoudre un problème n'est pas original, car dans notre vie quotidienne, on utilise souvent les connaissances acquises et nos expériences passées pour trouver des solutions à de nouvelles situations [MAL 96].

## 2. Approche du raisonnement à partir de cas

Le raisonnement à partir de cas est un des types de raisonnement en IA, dans le domaine de l'apprentissage automatique. Reasonner à partir de cas signifie se remémorer des situations passées, similaires à la situation courante et utiliser ces situations pour aider à résoudre la situation courante. Le raisonnement à partir de cas est une forme de raisonnement par analogie. L'analogie proprement dite recherche les relations de cause à effet dans les situations passées pour les transposer à la situation courante ainsi que les ressemblances entre les situations passées et la situation courante. Le raisonnement à partir de cas recherche seulement les ressemblances ou les relations de proximité entre les situations passées et la situation courante. Le RàPC envisage le raisonnement comme un processus de remémoration d'un petit ensemble de situations concrètes : les cas. Il fonde ses décisions sur la comparaison de la nouvelle situation (cas cible) avec les anciennes (cas sources). Le principe général du RàPC consiste à traiter un nouveau problème (cas cible) en se remémorant des expériences passées voisines (cas de référence). Ce type de raisonnement repose sur l'hypothèse suivante : si une expérience passée et la nouvelle situation sont suffisamment similaires, alors tout ce qui peut être expliqué ou appliqué à l'expérience

passée (base de cas) reste valide si on l'applique à la nouvelle situation qui représente le nouveau problème à résoudre [AAM 94].

## 2.1. Historique

Inspiré par les travaux de Minsky et Schank réalisés à la fin des années 70, Schank [SCH 82] formule pour la première fois le paradigme de raisonnement basé sur les cas. En effet, la théorie développée par Minsky [MIN 75] présente un réseau de nœuds et de relations entre ces nœuds ainsi que la notion de « frame (script, schéma) » qui correspond à une structure remémorée qui doit être adaptée pour correspondre à la réalité d'une nouvelle situation rencontrée. Cependant Schank doute de la flexibilité du raisonnement logique et d'une représentation des connaissances ordinaires sous une forme synthétique de propositions indépendamment vraies. Par conséquent, il reprend ces travaux et suppose que le processus de compréhension correspond à un processus d'explication qui s'applique d'une manière itérative [SCH 82]. D'ailleurs, Schank est considéré comme l'initiateur du terme « Case Based Reasoning ». Il introduit à travers le modèle de « mémoire dynamique » un degré de généralité varié connu sous le nom de « MOPS (Memory Organization Packets) » constituant un réseau dense d'expériences. De plus, l'auteur tente d'opérationnaliser le comportement humain et l'optimiser si possible. Dans ce cadre, Gebhardt et al. [GEB 97] définissent le raisonnement à partir d'expériences comme une façon naturelle de penser caractérisant la réflexion humaine sans doute plus encore que le raisonnement avec des règles.

A la fin des années 80, les recherches dans le domaine du RàPC ont réellement commencé à prendre forme et notamment avec les conférences « DARPA » organisées aux Etats-Unis en 1988 [KOL 88], avant de s'imposer en Europe avec la première conférence Européenne en 1993 à Kaiserslautern [RIC 93], puis avec la première conférence internationale à Lisbonne en 1995 [VEL 95].

## 2.2. Les origines des systèmes RàPC

Les origines du raisonnement à partir de cas sont la psychologie cognitive pour l'étude de la mémoire et le raisonnement par analogie. Le raisonnement à partir de cas complète le raisonnement par analogie par un mécanisme de mémorisation et d'extraction des expériences. C'est aussi une analogie intra-domaine, conçue pour une tâche bien précise. Par les mécanismes qu'il met en œuvre, notamment ceux liés à l'exploitation de la base de cas, le RàPC entretient aussi des relations avec d'autres formes de raisonnement, comme la classification et la catégorisation [HAT 91].

### 2.2.1. Théorie de la mémoire

Plusieurs théories de la mémoire ont successivement dominé dans les systèmes RàPC. La dernière étant la théorie de la mémoire dynamique de Schank [SCH 77]. Elle a donné lieu aux premiers systèmes de raisonnement à partir de cas [KOL 93] : la théorie de Lindsay, la théorie de la mémoire épisodique, la mémoire conceptuelle et la théorie de la mémoire dynamique : selon laquelle les processus cognitifs de compréhension, de mémorisation et d'apprentissage utilisent une même structure de mémoire. Cette structure, les "memory organization packets" (ou MOPS) est représentée à l'aide de schémas de représentation de connaissance tels que les graphes conceptuels et les scripts.

### 2.2.2. Le raisonnement par analogie

Par définition, l'analogie désigne « *rapport, similitude partielle d'une chose avec une autre* ». De nombreux exemples d'analogie existent dans notre entourage : l'analogie atome/système solaire, l'analogie courant électrique/cours d'eau. Le raisonnement par analogie consiste à avoir recours à un élément mieux connu pour inférer des informations sur un élément qui l'est moins. Il implique notamment l'évaluation de la ressemblance entre entités et leur mémorisation en vue de leur réutilisation [KOL 93].

#### 2.2.2.1. Finalités du raisonnement par analogie

Le raisonnement par analogie est reconnu être très utilisé par l'être humain. Dans la vie quotidienne, face à une situation donnée l'expérience d'une situation semblable peut s'avérer très utile. En intelligence artificielle, les systèmes mettent en œuvre ce raisonnement à diverses fins : pour la compréhension du langage naturel, la planification, etc.

Le raisonnement par analogie a pour objectif l'inférence d'informations sur une situation (la cible) à partir de la description d'une situation dans laquelle ces informations sont connues (la source).

Les deux composantes essentielles de l'analogie sont la mise en évidence des caractéristiques ou propriétés communes des situations et la détermination des relations intra- ou inter-domaines. En raisonnement par analogie, la ressemblance entre situations est basée souvent sur des critères syntaxiques (le raisonnement à partir de cas au contraire compare plutôt des ensembles de descripteurs) [KOL 93].

Il existe principalement deux contextes d'utilisation du raisonnement par analogie notamment dans le cadre de la résolution de problèmes :

- La résolution du problème s'annonce particulièrement complexe et longue. L'utilisation de la solution d'un problème similaire déjà résolu permet d'accélérer le processus de résolution,
- Dans le domaine considéré, il n'existe pas de théorie qui permet de résoudre le problème posé. L'utilisation d'un problème similaire résolu s'avère être la seule issue.

La première analogie est qualifiée d'« heuristique », et la seconde d'« analogie-recours ».

### 2.2.2.2. Formalisation et terminologie

Le paradigme d'analogie définit l'analogie comme la mise en œuvre d'un mécanisme de mise en correspondance ou de projection, entre des structures afin de transposer des connaissances d'un univers de base vers un univers cible, en fonction d'un certain point de vue pouvant correspondre à un but à atteindre ou un problème à résoudre [KOL 93].

Ce paradigme manipule deux entités appartenant chacune à un univers pouvant être réorganisé selon le point de vue adopté. L'une des deux entités est utilisée pour inférer des connaissances sur la seconde, elle appartient à l'univers de base, l'autre appartient à l'univers cible. Dans la description des mécanismes du raisonnement par analogie, ces deux entités sont appelées respectivement « base » et « cible ».

Un problème d'analogie s'exprime selon l'expression : « *D est à C ce que B est à A. Connaissant A, B et C, que vaut D ?* ». Le principe d'analogie est souvent schématisé par « le carré d'analogie » (Fig.1.1):

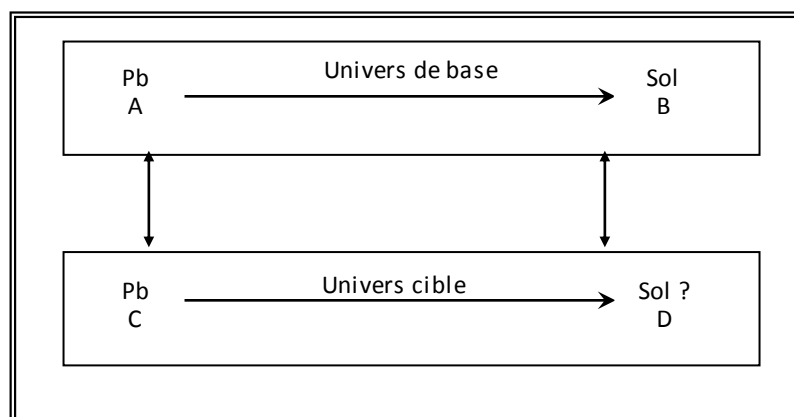


Fig.1.1. Carré d'analogie

Appliqué à la résolution de problème, ce principe a donné lieu à une définition plus précise [SMA 94]: « *La résolution de problèmes par analogie consiste à transférer de la connaissance à partir d'épisodes précédents de résolution, aux nouveaux problèmes qui partagent des aspects significatifs de l'expérience précédente et à utiliser les connaissances transférées pour construire des solutions pour les nouveaux problèmes* ».

Par rapport au carré d'analogie, A et B définissent un épisode de résolution, par un problème (A) et sa solution (B). C représente le problème à résoudre et D la solution recherchée. Carbonell propose deux approches de résolution de problème par analogie [SMA 94] :

- **Analogie par transformation** : Elle tente de réutiliser, avec des modifications, une solution précédemment trouvée pour un problème similaire.

Si  $d1$  représente la différence entre A et C, il s'agit de déterminer  $d2$ , la différence entre B et D afin d'en déduire D en propageant  $d2$  dans B. Le carré d'analogie devient alors (Fig.1.2) :

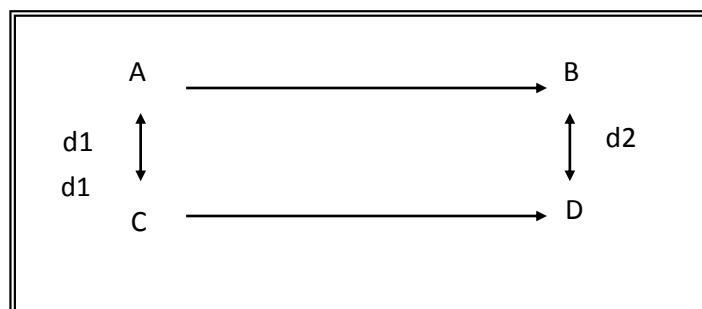


Fig.1.2. Analogie par transformation

- **Analogie par dérivation** : Il s'agit d'adapter le processus (ou la dérivation) ayant abouti à la solution d'un problème précédent pour construire la solution d'un problème similaire. Si  $S$  est la similarité entre A et C et  $P$  la méthode de construction de la solution B à partir du problème A. La résolution du problème C se fait alors par le calcul d'une méthode  $p'$  à partir de  $P$  en utilisant  $S$  ( $S$  sert à reconnaître les éléments de  $P$  qui sont encore valables dans la nouvelle situation C) (Fig.1.3) :

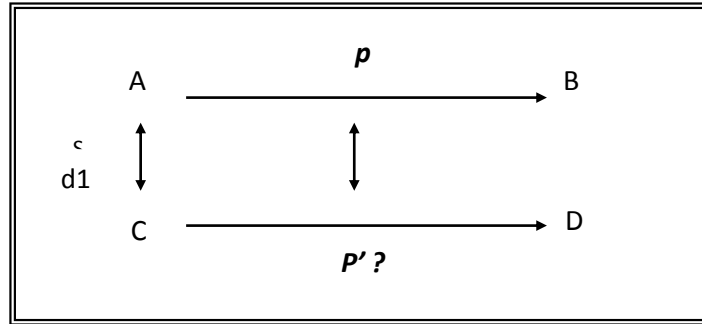


Fig.1.3. Analogie par dérivation

### 2.3. Représentation d'un cas

Un cas dans une base de cas représente différents types de connaissances qui peuvent être stockées selon différents formats de représentations. Un cas est un ensemble de problèmes se note « pb » et la solution de ce problème est alors codifiée « sol (pb) » :  $cas=(pb, sol (pb))$ .

Un « cas source » est un cas dont on va s'inspirer pour résoudre un nouveau problème et l'on nommera le « cas cible ». La codification est alors la suivante :

$$\text{Cas-source} = (\text{source}, \text{sol}(\text{source}))$$

$$\text{Cas-cible} = (\text{cible}, \text{sol}(\text{cible}))$$

Dans chacun de ces types de systèmes RàPC, un cas peut être représenté différemment. Selon les applications du RàPC, un cas, son problème et sa solution, sont décrits par un ensemble de descripteurs. Un descripteur est tout les informations qui nous permettent de décrire le problème. Le descripteur « d » est caractérisé par une paire  $d=(a,v)$ , où « a » est un attribut défini par un nom et « v » est la valeur qui lui est associée [GEB 97]. Un attribut représente une caractéristique du domaine applicatif qui peut être numérique ou symbolique.

Nous pouvons écrire un cas comme suit :

- $Source = \{d_1^S \dots d_n^S\}$  où  $d_i^S$  est un descripteur du problème source.
- $Sol(source) = \{D_1^S \dots D_n^S\}$  où  $D_i^S$  est un descripteur de la solution source.
- $Cible = \{d_1^C \dots d_n^C\}$  où  $d_i^C$  est un descripteur du problème cible.
- $Sol(cible) = \{D_1^C \dots D_n^C\}$  où  $D_i^C$  est un descripteur de la solution cible.

### 3. Composantes d'un système de Raisonnement à Partir de Cas

Un système RàPC est une combinaison de processus et de connaissances (“knowledge containers”) qui permettent de préserver et d’exploiter les expériences passées. Pour simplifier la présentation, nous nous appuyons sur le modèle générique présenté dans la figure (Fig.1.4) les principaux processus dont la recherche (“retrieval”), l’adaptation (“reuse”), la maintenance (“retain”) et la construction (“authoring”). Les structures de connaissances sont : le vocabulaire d’indexation, la base de cas, les métriques de similarité et les connaissances d’adaptation [HAJ 04].

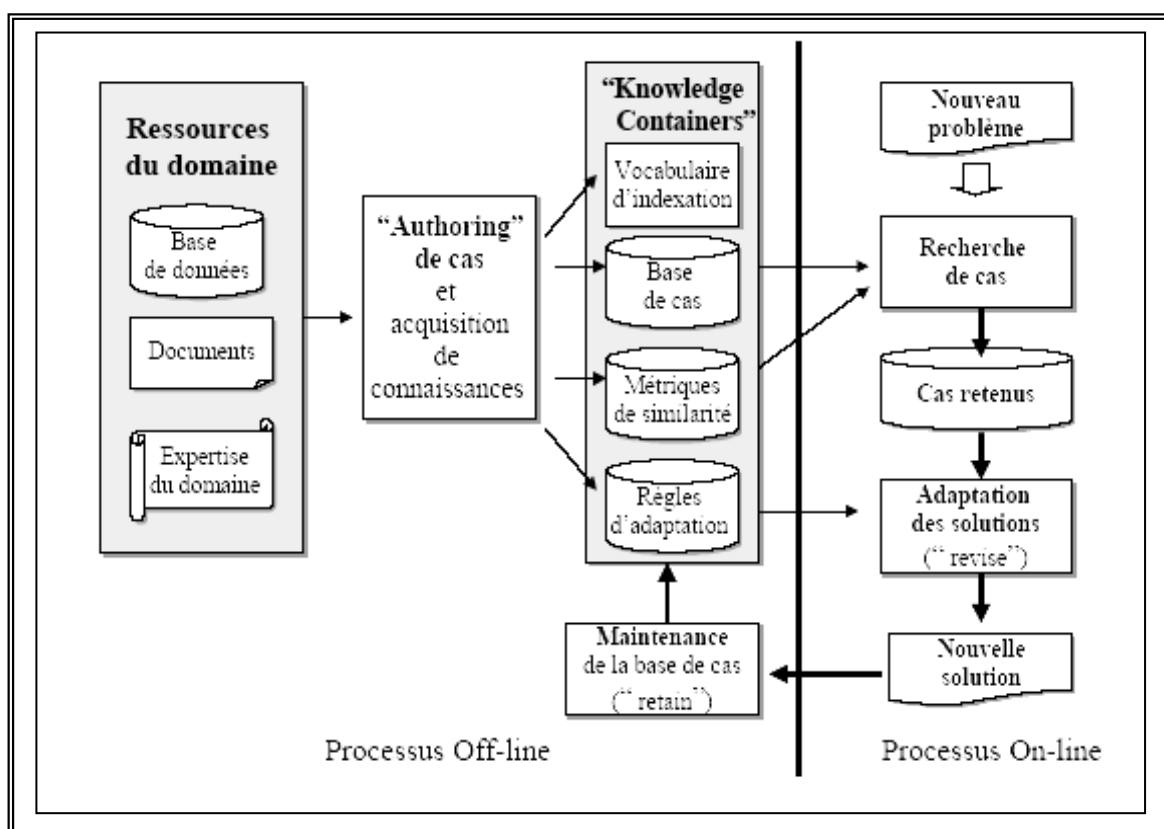


Fig.1.4. Modèle générique d'un système RàPC [AFO 04]

#### 3.1. Processus

Le souci fondamental des systèmes RàPC est d'éviter de reproduire les erreurs passées, et de faciliter l'acquisition des connaissances. Le RàPC doit donc se rappeler des cas pertinents de mémoire, puis à partir de la liste des cas retrouvés à l'étape précédente, sélectionner les cas les plus promoteurs en utilisant les mesures de similarités et construire une solution ou une interprétation pour le nouveau cas. Une solution est élaborée en adaptant les anciennes solutions. Ensuite, le RàPC doit tester et critiquer la sortie de l'étape précédente et proposer des contres exemples. Puis il évalue et analyse les résultats dans le



monde réel pour enfin mettre à jour la mémoire en stockant et en indexant le nouveau cas [AFO 04] [ARM 09].

Lorsqu'un nouveau cas est à résoudre, il est intégré dans un cas cible où la partie solution est inconnue et doit être apportée par le raisonnement. Les cas sources représentent des expériences passées stockées dans une mémoire. Les cas sources et cibles ont le même formalisme de représentation. L'objectif du raisonnement en cinq phases est de transférer des enseignements pertinents des cas sources pour élaborer la solution du cas cible.

La figure (Fig.1.5) résume clairement le cycle d'un RàPC :

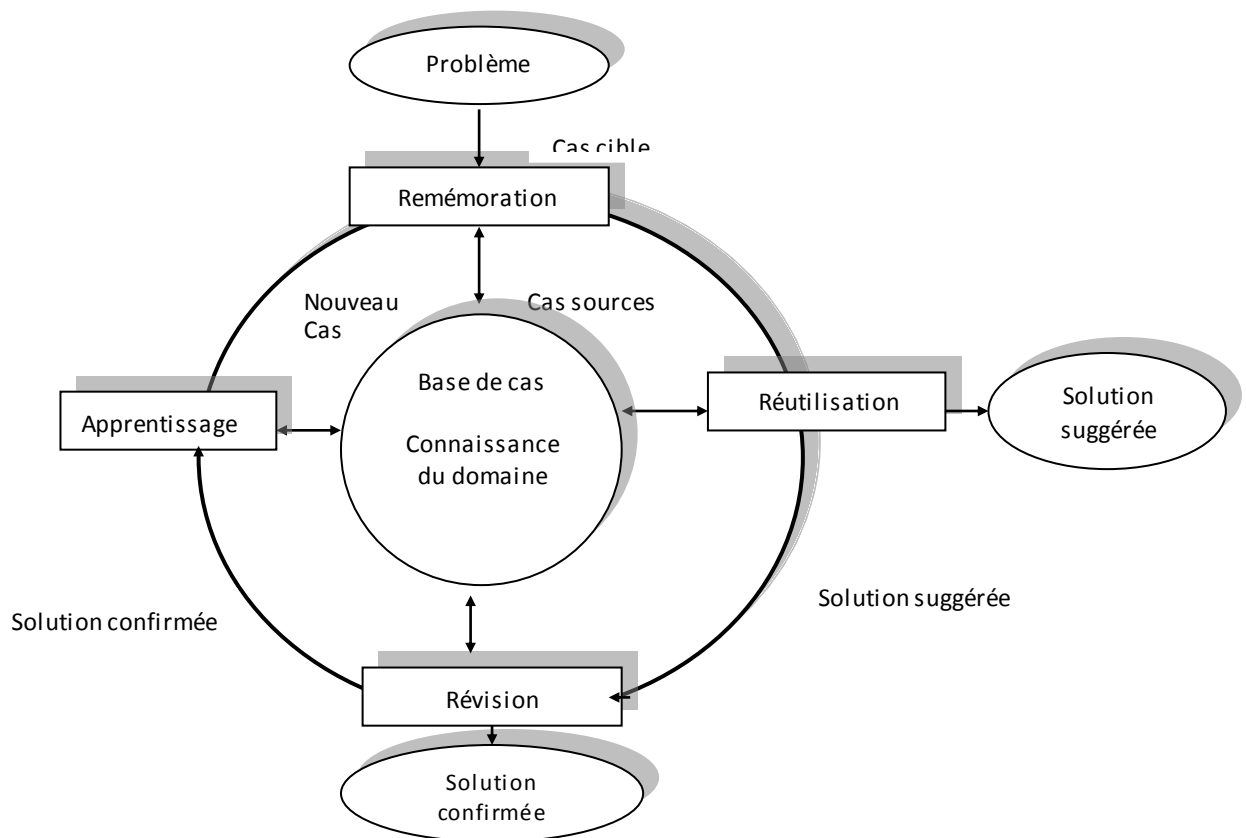


Fig.1.5. Cycle du raisonnement à partir de cas [AFO 04]

### 3.1.1. Remémoration

La phase de remémoration est une étape importante dans un système RàPC. L'une des hypothèses majeures dans le RàPC est de dire que des expériences similaires peuvent guider de futurs raisonnements, des résolutions de problèmes et permettre un apprentissage de la base de cas. La remémoration dépend essentiellement de la représentation de cas, de leur indexation et de leur organisation de la base de cas. Cette phase est donc le processus qui

permet de retrouver des cas sources similaires au cas cible. Les cas sources similaires sont des cas qui ont besoin de moins d'adaptation [SOL 04] [HAO 09].

### 3.1.1.1. Les techniques de recherche des cas similaires

#### - *Mesure de similarité*

Ces mesures de similarité cherchent des correspondances entre descripteurs de ces cas qui sont ensuite retrouvés à l'aide d'un algorithme de recherche. L'objectif de ces mesures de similarité est de retrouver le cas de la base de cas similaire au problème actuel au sens qu'il est facilement adaptable à ce nouveau problème. Il existe différentes méthodes pour calculer la similarité ou la dissimilarité. La distance de Minkowski est la plus connue et est souvent utilisée. Il s'agit de calculer la distance relative entre le cas cible et le cas source [FUC 08][PAL 04][ARM 09].

$$Sim(C, S) = \left[ \sum_{i=1}^n w_i * |C_i - S_i|^p \right]^{1/p}$$

$N$  : est le nombre d'attribut

$C$  : est le cas cible

$C_i$  : est le descripteur de cas cible

$S$  : est le cas source

$S_i$  : est le descripteur du cas source

$W$  : est le poids de l'attribut  $i$

En fonction du paramètre  $p$  :

$P=1$  distance de Manhattan

$P=2$  distance euclidienne

$P=\infty$  distance de Chebychev ( $\text{Max}|C_i - S_i|$ )

Plusieurs techniques de similarité existent dans la phase de remémoration : *K Plus Proches Voisins (KPPV)*, les *approches inductives*, la *logique floue* ou encore les *réseaux neurones*.

Nous décrivons par la suite les deux approches plus utilisées dans ce domaine : *L'approche K Plus Proches Voisins* ainsi que *les approches inductives*.

#### ***L'approche K Plus Proches Voisins***

La technique K Plus Proches Voisins « *KPPV* » (plus connus en Anglais sous le nom *K-Nearest Neighbors (K-NN)* [WEI 91]) est la technique la plus utilisée dans les RàPC.

Cette technique est établie selon la mesure de Minskowski. Elle est basée sur les poids des descripteurs  $w_i$  qui évaluent la similarité entre les cas sources de la base de cas et le cas cible. Le poids permet alors de modifier l'importance d'un descripteur par rapport à un autre descripteur. La partie la plus délicate dans cette technique est de définir de poids des descripteurs. Dans l'équation 1,  $w$  est le poids du descripteur,  $sim$  est la fonction de similarité, et  $d_i^C$  et  $d_i^S$  sont des valeurs du descripteur  $i$  dans le cas cible et respectivement dans le cas source.

$$Similarité(C,S) = \frac{\sum_{i=1}^n w_i \times sim(d_i^C, d_i^S)}{\sum_{i=1}^n w_i} \quad (1)$$

### ***Les approches inductives***

Les approches inductives déterminent quels sont les descripteurs qui apportent un meilleur effet pour une discrimination des cas. Les algorithmes d'Induction, tels que ID3 [QUI 86] et CART (Classification And Regression Trees), permettent de construire des arbres de décision à partir des cas de la base de cas. Ces algorithmes d'induction divisent les cas en grappes (clusters). Chaque cluster contient des cas similaires. Les techniques d'Induction sont utilisées majoritairement comme des classificateurs pour regrouper des cas similaires. Ils déterminent quels descripteurs sont à même de mieux distinguer les cas et de générer une structure de l'arbre de décision pour organiser le cas de la base de cas.

#### **3.1.2. Adaptation ou réutilisation**

Suite à la sélection de cas lors de la phase de recherche, le système RàPC aide l'utilisateur à modifier et à réutiliser les solutions de ces cas pour résoudre son problème courant. En général, on retrouve deux approches d'adaptation de cas [BUI 04] [DAQ 04] [HAO 09]:

- ✓ *l'approche transformationnelle (ou structurelle)* : on obtient une nouvelle solution en modifiant des solutions antécédentes et en les réorientant afin de satisfaire le nouveau problème.
- ✓ *l'approche générative (ou dérivationnelle)* : on garde, pour chaque cas passé, une trace des étapes qui ont permis de générer la solution. Pour un nouveau problème, une nouvelle solution est générée en appliquant l'une de ces d'étapes.

Peu de systèmes RàPC font de l'adaptation complètement automatique. Pour la plupart des systèmes, une intervention humaine est nécessaire pour générer partiellement ou complètement une solution à partir d'exemples. Le degré d'intervention humaine dépend des

bénéfices en terme de qualité de solution que peut apporter l'automatisation de la phase d'adaptation [HAM 89].

### 3.1.3. Maintenance ou révision

Durant le cycle de vie d'un système CBR, les concepteurs doivent préconiser certaines stratégies pour intégrer de nouvelles solutions dans la base de cas et pour modifier les structures du système RàPC pour en optimiser les performances. Une stratégie simple est d'insérer tout nouveau cas dans la base. Mais d'autres stratégies visent à apporter des modifications à la structuration de la base de cas (e.g. indexation) pour en faciliter l'exploitation. On peut également altérer les cas en modifiant leurs attributs et leur importance relative. Cet aspect de recherche est actuellement l'un des plus actifs du domaine RàPC [LEA 01] [SOL 04].

Cette étape a pour but la validation de la solution produite. La confrontation de cette solution au monde réel détecte les erreurs des deux étapes précédentes. Elle est généralement réalisée dans une boucle évaluation / correction.

### 3.1.4. Construction ou apprentissage

Ce processus, en amont des activités de résolution de problèmes du système RàPC, sous-tend la structuration initiale de la base de cas et des autres connaissances du système à partir de différentes ressources tels des documents, bases de données ou transcriptions d'interviews avec des praticiens du domaine. Ce processus, souvent effectué manuellement par le concepteur du système, se prête moins bien à l'automatisation car il nécessite une connaissance du cadre applicatif pour guider, entre autre, la sélection du vocabulaire d'indexation et la définition des métriques de similarités [AAM 98].

## 3.2. Connaissances dans un système RàPC

Richter [RIC 98] définit les systèmes de RàPC comme des systèmes à base de connaissances (SBC). Ces systèmes exploitent quatre catégories de connaissances ("knowledge containers") distinctes à savoir :

- ✓ *vocabulaire d'indexation* : un ensemble d'attributs ou de traits ("features") qui caractérisent la description de problèmes et de solutions du domaine. Ces attributs sont utilisés pour construire la base de cas et jouent un rôle important lors de la phase de recherche.

- ✓ *base de cas* : l'ensemble des expériences structurées qui seront exploitées par les phases de recherche, d'adaptation et de maintenance.
- ✓ *mesures de similarité* : des fonctions pour évaluer la similarité entre deux ou plusieurs cas. Ces mesures sont définies en fonction des traits et sont utilisées pour la recherche dans la base de cas.
- ✓ *connaissances d'adaptation* : des heuristiques du domaine, habituellement sous forme de règle traits permettant de modifier les solutions et d'évaluer leur applicabilité à de nouvelles situations.

## 4. Organisation de la mémoire de cas

Les processus de mémorisation et de remémoration sont fortement liés à la façon d'organiser les cas dans la base. Dans cette section, nous allons aborder différentes méthodes d'organisation des cas en mémoire. Elles se résument en deux catégories principales [KOL 93] [AAM 94]:

- *La mémoire plate* : il s'agit de mémoriser tous les cas dans une liste séquentielle. Nous détaillons ce type de mémoire dans la section suivante.
- *La mémoire hiérarchique* : lorsque la mémoire des cas est large, il y a une nécessité d'organiser les cas hiérarchiquement. Ceci permet de simplifier la remémoration. Par la suite, nous présentons deux approches pour l'organisation hiérarchique des cas en mémoire : *les réseaux à trait partagés* et *les arbres de discriminations*.

### 4.1. Organisation plate

Les cas sont stockés séquentiellement dans une simple liste ou fichier, c'est la plus simple structure à imaginer pour une mémoire. Les avantages d'une telle structure est que pendant la remémoration, tous les cas existant dans la mémoire sont testés, ceci garantie une remémoration précise qui dépend de la quantité de la fonction d'appariement. De plus, la mémorisation n'est pas coûteuse, il suffit d'ajouter le nouveau cas à la fin du fichier.

L'inconvénient majeur est le temps de remémoration qui augmente linéairement avec la taille de la mémoire. Il existe plusieurs variantes pour l'organisation en mémoire plate [MAL 96]:

- *Indexation superficielle* : l'indexation se fait dans un seul niveau, chaque descripteur (attribut-valeur) choisi comme étiquette pointe vers les cas correspondants (qui contiennent ce descripteur dans leur représentation).

Pendant la phase de remémoration, les cas qui sont pointés par ce descripteur sont sélectionnés, et ensuite la fonction d'appariement est appliquée à ce groupe des cas, et non à tous les cas. Ceci fonctionne très bien quand ces descripteurs permettent d'extraire un petit nombre de cas quand les étiquettes sont suffisamment descriptives.

- *Partitionnement de la mémoire* : le système doit reconnaître à quelles partitions une nouvelle situation appartient. La fonction d'appariement est appliquée seulement aux cas appartenant aux partitions sélectionnées.
- *Extraction parallèle* : la fonction d'appariement est d'appliquée d'une manière parallèle à tous les cas qui existent dans la mémoire.

#### **4.2. Réseaux à caractéristiques partagées**

Les réseaux à caractéristiques partagées regroupent les cas présentant des similarités dans un même cluster. Les hiérarchies sont formées lorsque les clusters sont subdivisés en sous clusters. Cette organisation offre l'avantage de mieux partitionner la base de cas et rend la recherche plus efficace. Mais quelques inconvénients, telle que la complexité lors de l'ajout d'un cas, la difficulté de maintenir l'optimalité du réseau lors de l'ajout. Un espace supplémentaire est requis pour l'organisation.

Plusieurs réseaux avec des priorités différentes seraient nécessaires pour augmenter la précision de la recherche.

#### **4.3. Réseaux de discrimination**

Le regroupement effectué dans les réseaux à caractéristiques partagées conduit à une discrimination en second lieu. Dans ces réseaux, chaque nœud interne est une question qui départage selon la réponse des cas de la base. Les questions les plus importantes sont posées en premier. Ce type d'organisation, en plus des avantages des réseaux à caractéristiques partagées, rend intuitive la compréhension de la connexion entre indices et l'organisation du réseau. Les inconvénients sont ceux des réseaux à caractéristiques partagées, plus le problème de traitement des informations manquantes.

#### 4.4. Réseaux redondants de discrimination

Les réseaux redondants de discrimination fournissent une réponse au problème des informations manquantes. Ils organisent les cas en utilisant différents réseaux de discriminations, chacun avec un ordre différent des questions. Une recherche se fait en parallèle sur les différents réseaux. Si dans l'un des réseaux une question n'a pas de réponse, on y abandonne la recherche. Au moins, l'un des réseaux retrouvera le cas qui s'apparie s'il existe. L'inconvénient majeur d'une telle organisation est la complexité de sa mise en œuvre.

### 5. Exemple des systèmes RàPC

#### 5.1 Exemple 1 : Calcul du facteur de risque d'une voiture pour une voiture

On veut déterminer le facteur de risque (nombre entier compris entre -3 et 3) pour une voiture donnée (cas cible) en fonction des assignations déjà effectuées sur d'autres.

voitures (cas sources). L'approche CBR repose sur l'utilisation d'une base de cas dans laquelle un cas associe la description des caractéristiques d'une voiture (problème formé d'indices) au facteur de risque (solution). On propose alors d'effectuer un raisonnement en trois étapes dans lequel la phase de révision n'est pas utilisée (cf. Figure 8).

- La phase de recherche permet d'identifier un ensemble de cas dont la description est proche de celle de la voiture courante pour laquelle il s'agit de déterminer le facteur de risque. La stratégie de recherche mise en œuvre pour les évaluer est la sélection par plus proches voisins.
- La phase de réutilisation prend en compte l'ensemble des cas retrouvés pour calculer le facteur de risque, en effectuant une moyenne des facteurs de risque assignés aux cas sources pondérés par leur similarité.

Enfin, la phase d'apprentissage ajoute le nouveau cas à la mémoire, s'il n'existe pas déjà un cas trop proche du cas cible courant, et si la base de cas ne dépasse pas une taille donnée.

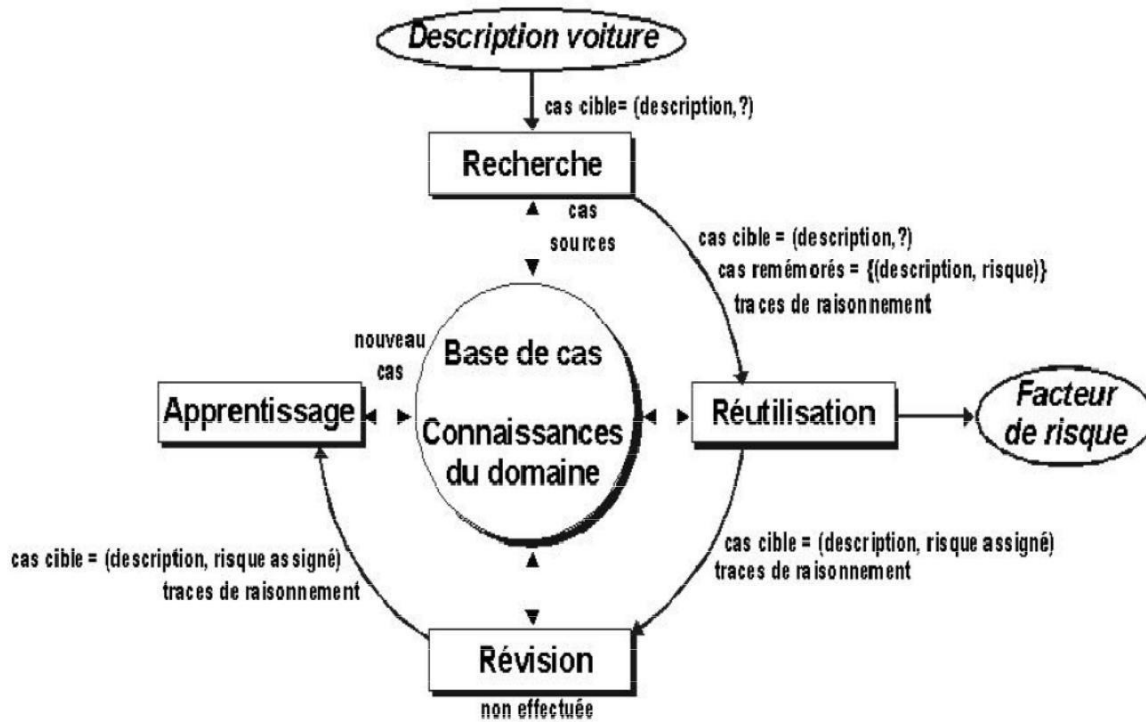


Figure 4.8: Calcul du facteur de risque pour une voiture

## 5.2. Exemple 2 : Système de recherche des prix

Supposons que l'on désire mettre en place un système de recherche des prix des appartements à Toulouse, en fonction de la zone où ils sont situés et du type d'appartements (T1, T2, ...)

Nous disposons des informations suivantes :



Cas	Zone	Ttpe	Prix ( euro)
1	Rangueil	T1	500
2	Rangueil	T3	900
3	Muret	T4	1000
4	Muret	T2	340
5	St Agne	T1	400
6	St Agne	T3	550
7	St Agne	T5	780
8	Minimes	T1	380
9	Minimes	T1	275
10	Minimes	T6	1500

Nous avons des cas identifiés par un ID\_Cas , possédant des caractéristiques qui sont la zone et le type, et comme solution nous avons le prix.

Nous allons illustrer comment utiliser un raisonnement à partir des cas pour réaliser le système voulu.

Indexation des cas : Afin de faciliter la recherche, nous allons classer les cas par zone : on obtient les classes  $C_1, \dots, C_n$ . Nous avons ainsi dans le cas de notre exemple les classes suivantes :

- C1 : zone Rangueil
- C2 : zone Muret
- C3: zone St Agne
- C4 : zone Minimes

Recherche des cas : lorsqu'un cas courant e est à résoudre, l'algorithme que nous allons utiliser est le suivant :

Si le nouveau cas appartient à la classe  $C_i$  :

1. Pour chaque cas  $e_j$  dans  $C_i$  on calcule la similarité par rapport à  $e$  grâce à la formule suivante :

$$\text{Sim}(e, e_j) = (\text{caractéristiques communes}) / (\text{caractéristiques communes} + \text{caractéristiques différentes})$$

2. On classe les cas par similarité calculée
3. On choisit les cas  $e_1, \dots, e_m$  tel que  $\text{sim}(e, e_k)$  est égale au maximum des similarités calculées
4.  $e_1, \dots, e_m$  sont les cas similaires à  $e$

Dans notre exemple si le nouveau cas est  $e = (\text{St Agne}, T5, ?)$ , la phase de recherche va produire les résultats suivants :

CAS RETIRES

CAS SIMILAIRES

$e=(5, \text{St Agne}, T1, 400) : \text{sim} = 0.5$

$e=(6, \text{St Agne}, T3, 550) : \text{sim} = 0.5$

$e=(7, \text{St Agne}, T5, 780) : \text{sim} = 1$

$e=(7, \text{St Agne}, T5, 780)$

Adaptation des cas : l'algorithme d'adaptation est le suivant :

1. On classe les cas similaires en fonction de leur solution : on obtient les classes  $I_1, \dots, I_n$
2. On compte le nombre de cas contenu dans chaque catégorie  $I_i$ , soit  $n_1, \dots, n_m$  et on calcule  $n_{\max} = \max \{ n_1, \dots, n_m \}$
3. Si  $n_i > n_j$ ,  $i$  différent de  $j$  alors la solution du nouveau cas est égale à la solution de la catégorie  $I_i$
4. S'il existe plusieurs catégories qui ont le nombre max, on prend la moyenne de leurs solutions

Dans le cas de notre exemple, la solution pour le nouveau cas sera 780 euro

## 6. Conclusion

Nous avons introduit des principes fondamentaux d'un système de raisonnement à partir de cas et des exemples sur des systèmes RàPC. Notons que la mémoire de cas joue un rôle très important dans le RàPC, plus elle s'accroît, et plus les temps de calcul sont longs. C'est pourquoi plusieurs techniques d'organisation de la mémoire ont été proposées.

## Bibliographie

- [AAM 94] Aamodt A. et Plaza E., Case Based Reasoning: Foundational issues, methodological variations, and system approaches. AI communications, IOS Press, Vol 7(1), pp. 39-59, 1994.
- [AAM 98] Aamodt A. et Langseth H., Integrating Bayesian Networks into Knowledge-Intensive CBR. In: AAI Workshop on Case-Based Reasoning Integrations, 1998. Technical Report WS-98-15. AAI Press, Menlo Park, ISBN 1-57735-068-5, pp. 1-6, 1998.
- [AFO 04] Afouba N., Kerbrat S. et Labarang Z., Rapport du projet d'intelligence artificielle : Le raisonnement à partir des cas : Définitions et principes de fonctionnement, Novembre 2004.
- [ARM 09] Armaghan N., Contribution à un système de retour de l'expérience basé sur le raisonnement à partir de cas conversationnel : application à la gestion des pannes de machines industrielles. Ecole doctorale Science et Ingénieur (RP2P),

- Université de Nancy, Mai 2009.
- [BUI 04]** Buist E., Les éléments fondamentaux du raisonnement à base de cas Les éléments fondamentaux du raisonnement à base de cas. pp. 1-46, Février 2004. <http://www.ericbuist.com/me/travaux/cbr.pdf>
- [DAQ 04]** D'Aquin M., Brachais S., Lieber J., et Amedeo N., 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. In Kanawati, R., Salotti, S., & Zehraoui, F. (Eds.), Actes du douzième atelier raisonnement à partir de cas, RàPC'04, pp. 41-52, 2004.
- [FUC 08]** Fuchs B., Raisonnement à Partir de Cas, In Renaud, J., Chabel Morello B, et Metta N, (éd), Retour et capitalisation d'expérience, outils et démarche. La Plaine Saint-Denis : AFNOR, pp. 184, 2008.
- [GEB 97]** Gebhardt F., Voß A., Gräther W. et Schmidt-Belz B., Reasoning With Complex Cases. Kluwer academic publishers, 1997.
- [HAJ 04]** HajSaid A., Distances sémantiques pour la comparaison des connaissances objets dans le cadre du raisonnement à partir de cas. Mémoire de DEA Informatique Théorique et Applications, Université de le Havre, pp. 25-40, 2004.
- [HAM 89]** Hammond K.J., Adaptation of Cases. In: Proceedings of DARPA Workshop on Case- Based Reasoning, pp. 88–89, Florida, June 1989.
- [HAO 09]** Haouchine M. K., Remémoration guidée par l'adaptation et maintenance des systèmes de diagnostic industriel par l'approche du raisonnement à partir de cas. L'UFR des Sciences et Techniques de l'Université de Franche-Comté, Septembre 2009.
- [HAT 91]** Haton J. P., Bouzid N., Charpillet F., Haton M. C. B., Laasri H., Marquis P., Mondot T. et Napoli A., Le raisonnement en Intelligence Artificielle, pp. 8-9, InterEditions, Paris, 1991.
- [KOL 88]** Kolodner J., Workshop on case-based Reasoning. Editor *DARPA 88*, Clearwater, Morgan Kaufmann, San Mateo, Florida, 1988.
- [KOL 93]** Kolodner J., Case-Based Reasoning. Morgan Kaufmann, San Mateo, Publishers, Inc, UCA, 1993.
- [LEA 01]** Leake D. B., Smyth B., Yang Q. et Wilson D., Special Issue on Maintaining Case- Based Reasoning Systems. Computational Intelligence, Vol 17(2), 2001.
- [MAL 96]** Malek M., Un modèle hybride de mémoire pour le raisonnement à partir de cas. Thèse de doctorat, Université Joseph Fourier, 30 octobre 1996.
- [MIN 75]** Minsky M., A framework for representing knowledge. In the psychology of Computer Vision, Editeur P.H. (Ed.) Winston, pp. 211-279, New York, McGraw Hill, 1975.
- [PAL 04]** Pal S. K. et Shiu, S.C.K., Foundation of Soft case-based reasoning, Wiley Series on Intelligent Systems IEEE, Albus, J.S., Meystel, A.M., Zadeh, L.A., Series

---

Editors, ISBN 0471086355, 9780471086352, 274 pages, 2004.

- [QUI 86]** Quinlan J., Induction of decision trees. Machine Learning, Vol 1, pp. 81-106, 1986.
- [QUI 86]** Quinlan J., Induction of decision trees. Machine Learning, Vol 1, pp. 81-106, 1986.
- [RIC 98]** Richter M.M., Introduction In Case-Based Reasoning Technology: From Foundations to Applications. Edited by M. Lena, B. Bartsch-Sporl, H. D. Burkhard, et S.Wess. Springer-Verlag, Berlin, pp. 1-15, 1998.
- [SCH 77]** Schank R. et Abelson R., Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA, 1977.
- [SCH 82]** Schank R.C., Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Cambridge University Press, New York, NY, 1982.
- [SMA 94]** Smail M., Raisonnement à base de cas pour une recherche évolutive d'information. Prototype CARBI-n.vers la définition d'un cadre d'acquisition de connaissance, Thèse de doctorat d'université, Unix. HERNI Pointcaré-Nancy I, Octobre 1994.
- [SOL 04]** Sollenborn M., Clustering and Case-Based Reasoning for User Stereotypes. Copyright C, ISSN 1651-9256, ISBN 91-88834-66-2, October 2004.
- [VEL 95]** Veloso M., Carbonell J., Pérez A., Borrajo D., Fink E. et Blythe J., Integrating Planning and Learning: The PRODIGY Architecture. Journal of Experimental and Theoretical Artificial Intelligence, Vol 7(1), pp. 81-120, 1995.
- [WEI 91]** Weiss S.M. et Kulikowski C.A., Computer systems that learn. Artificial intelligence ISSN 0004-3702, Morgan-Kaufmann, 1991, 1993, Vol 62(2), pp. 363-378, 1991.