Raisonnement à Partir de Cas

Sommaire:

- Introduction.
- Historique du RàPC.
- Qu’est ce qu’un cas?
- Ontologie.
- Qu’est ce qu’une base de cas.
- Choix d’un cas source dans la base de cas.
- Principe du RàPC.
- Conclusion

Introduction

Pour résoudre les problèmes de la vie quotidienne, nous faisons naturellement appel à notre expérience. Nous nous remémorons les situations semblables déjà rencontrées. Puis nous les comparons à la situation actuelle pour construire une nouvelle solution qui, à son tour, s’ajoutera à notre expérience.

Introduction

- Le raisonnement a partir de cas est un paradigme de l'intelligence artificielle qui consiste a utiliser les solutions de problèmes passes, déjà résolus, dans le but de résoudre de nouveaux problèmes. La réutilisation peut consister en la simple reapplication de la solution précédente, ou bien en l'application d'une solution ⩽ adaptée ⩾ pour mieux répondre aux spécificités du problème.

Historique du RàPC

- Marvin Minsky (psychologue) : grand pere fondateur de l'IA (decede en janvier 2016), tres provocateur dans sa facon de voir l'IA. Sa premiere grande contribution : concept de *frame.*
- Selon Minsky, notre pensee est organisee en schema. *"Pour toutes les tâches légèrement complexes de notre quotidien, on a des schémas." → Aller au restaurant, attendre d'être rentré, être placé... Chez Bob*
- Grand ⩽ revolutionnaire ⩾ : il a une approche peu conventionnelle de l'IA et pousse les chercheurs et ses collaborateurs a aller ⩽ au-dela ⩾ des evidences.

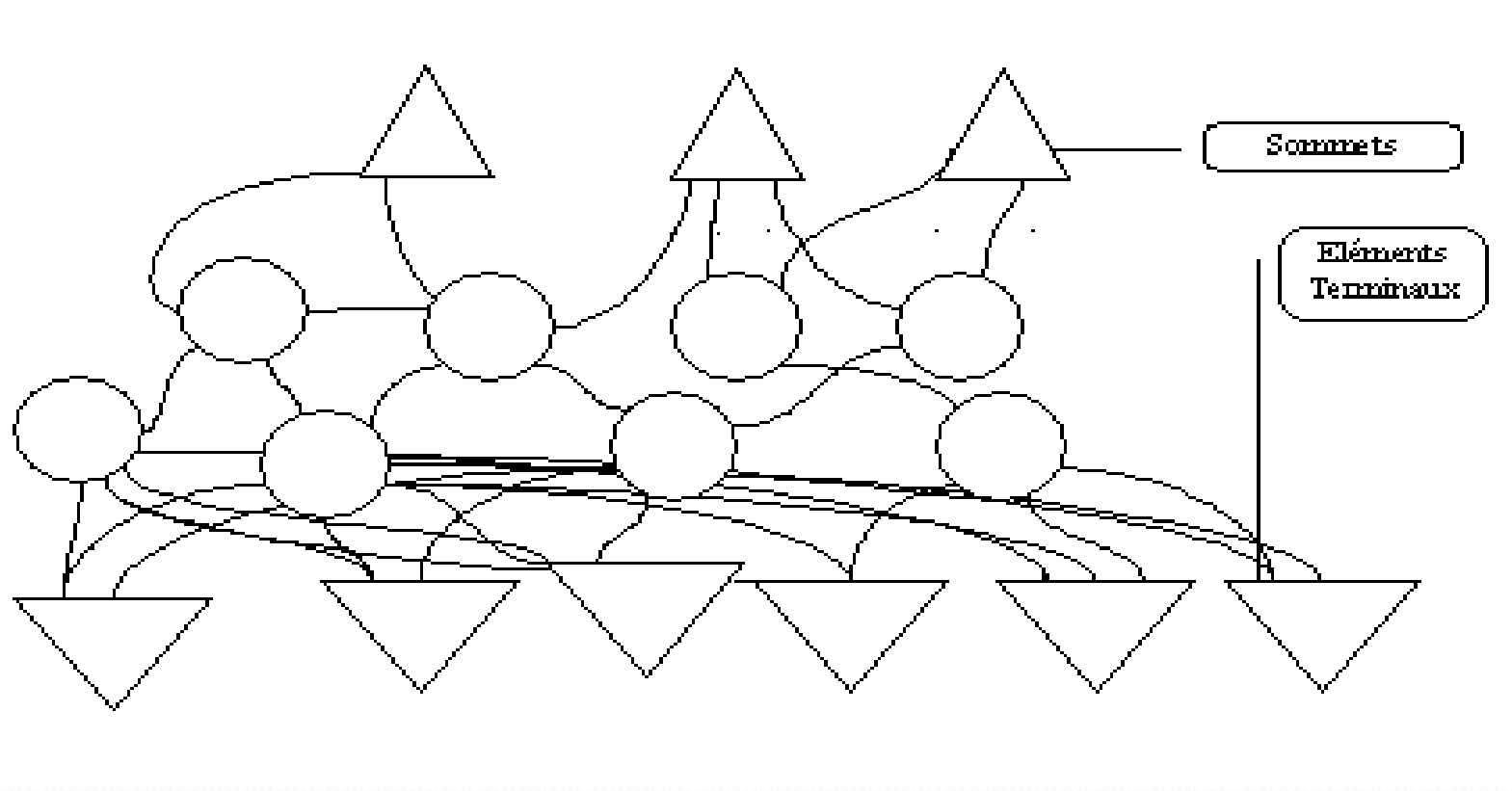
Historique du RàPC

- Roger Schank (specialiste, et precurseur en Intelligence Artificielle) invente les Memory Organisation Packets. Son hypothese va plus loin que celle de Minsky : on organise les schemas par paquet, et quand on a besoin de parcourir le schema, on ouvre le paquet et on l’explore (travaux realises entre 1975 et 1982).

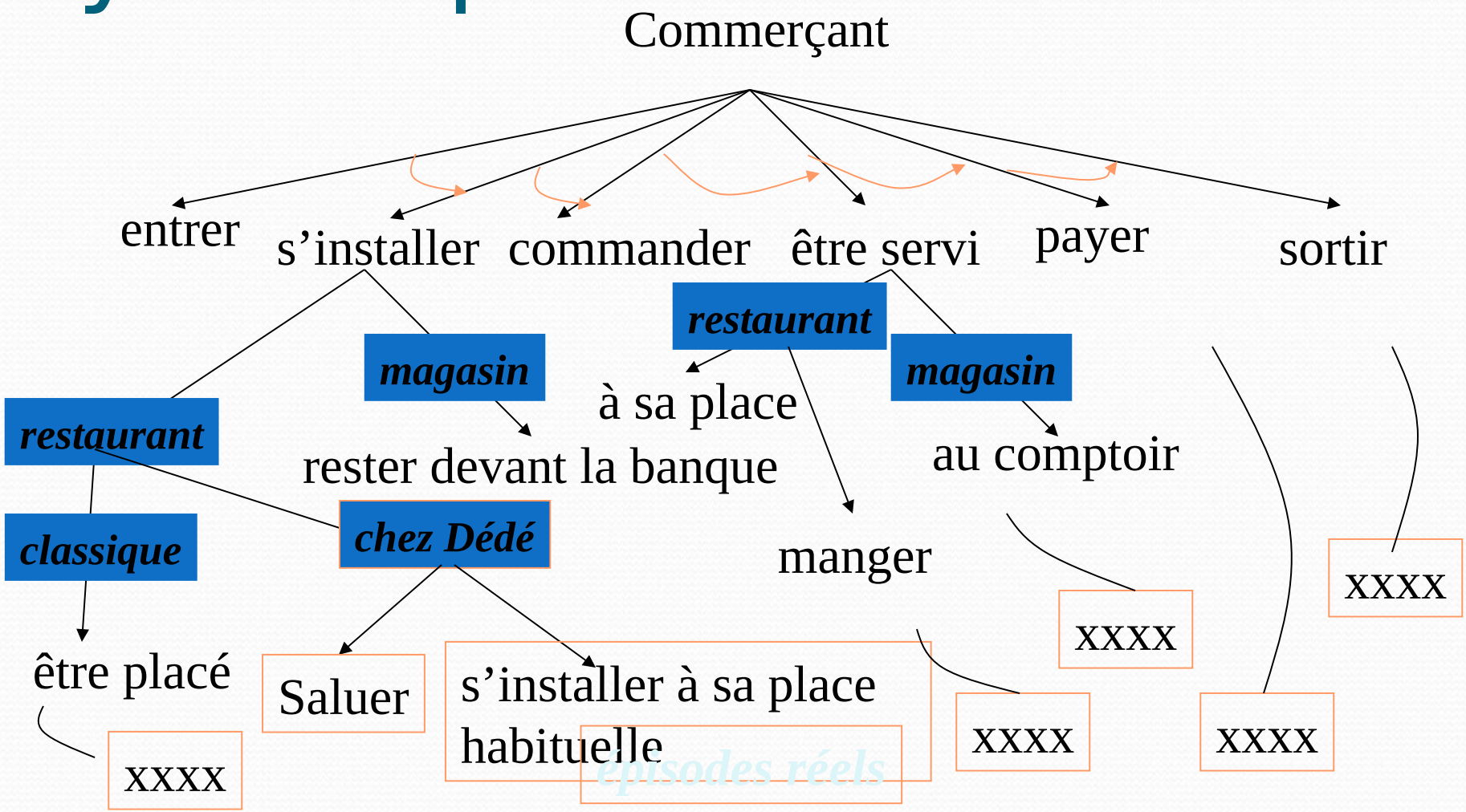
Historique du RàPC

- Janet L. Kolodner, auteur de *“Inside Case-Based Reasoning” en 1983, introduit* pour la première fois le terme de ⩽ *case-based reasoning » (raisonnement a partir* de cas). Son ouvrage est un ouvrage de synthèse des travaux de l’epoque, dans lequel elle s’appuie fortement sur les travaux de Minsky et Schank.
- En 1994, Aamodt et Plaza publient un article désormais célèbre dans lequel ils proposent un ⩽ cycle ⩾ du raisonnement a partir de cas compose de 4 etapes: (retrive, reuse, revise and retain). C’est le cycle le plus souvent utilise pour expliquer comment fonctionne le RaPC.

Quelques concepts importants



Des scripts à la mémoire dynamique



Des scripts à la mémoire dynamique

- Pour pouvoir resoudre un probleme, il est necessaire d'etre capable de decrire la situation en terme d'indicateurs, de fouiller la memoire a l'aide de descripteurs pour trouver des schemas correspondants au probleme et d'essayer d'appliquer ces schema au probleme. Si le schema n'est plus applicable, il est necessaire d'adapter le schema.

RàPC

- Depuis la publication du livre de Janet Kolodner, la communaute de recherche sur le raisonnement a partir de cas s’est structuree : il existe plusieurs ressources disponibles, gerees par une communaute rassemblee autour d’une conference annuelle (ICCBR) et d’un ensemble d’activites .

- Dans les annees 1990, l'IA a rencontre un grand succes industriel, en grande partie grace aux systemes experts (systemes a base de regles). Malheureusement, a cause des difficultes a modeliser les connaissances, mais surtout a maintenir le systemes, l'engouement s'est vite eteint. Les systemes experts requierent une modelisation relativement exhaustive du monde sur lequel on souhaite raisonner. Or, tout modeliser est loin d'etre possible.
- A l'inverse, dans le cas du raisonnement a partir de cas, on memorise les experiences de resolution de problemes au fur et a mesure qu'on les rencontre (diagnostic, planification, aide a la decision, conception, etc.). Certes, au demarrage, le systeme est moins performant, mais il gagne en competences au fur et a mesure, et il est moins difficile a maintenir. De plus, le RaPC est tres adapte a de nombreux domaines (listes juste avant), et correspond bien a une de nos facon de raisonner… beaucoup de gens se l'approprient donc assez facilement.

Qu’est ce qu’un cas?

- Un cas est la description d'un épisode de résolution de problème.

- Il peut prendre des formes très diverses selon la nature de la tâche : diagnostic, planification, aide à la décision, conception, etc.

Descripteurs de cas:

- Un cas est l'association d'un problème et de la solution de ce problème : ***cas=(pb,Sol(pb))***.
- Un **cas source** est un cas dont on va s'inspirer pour résoudre un nouveau cas appelé: **cas cible**.
- Un cas source s'écrit : ***cas-source=(source,Sol(source))*** et un cas cible s'écrit ***cas-cible=(cible,Sol(cible))***.
- Un cas, son problème et sa solution sont décrits par un ensemble de descripteurs.
- Un descripteur d est défini par une paire d=(a,v) où a est un attribut et v la valeur qui lui est associée dans ce cas.

.

Exemple(1):

Cas de détermination de conditions de vente d'un appartement:

Partie problème

d^s_1 = Surface de l'appartement (un réel)

d^s_2 = Localisation de l'appartement (une structure de données)

d^s_3 = Etat de l'appartement (une liste de défauts)

Partie solution

D^c_1 = Prix de vente de l'appartement (un réel)

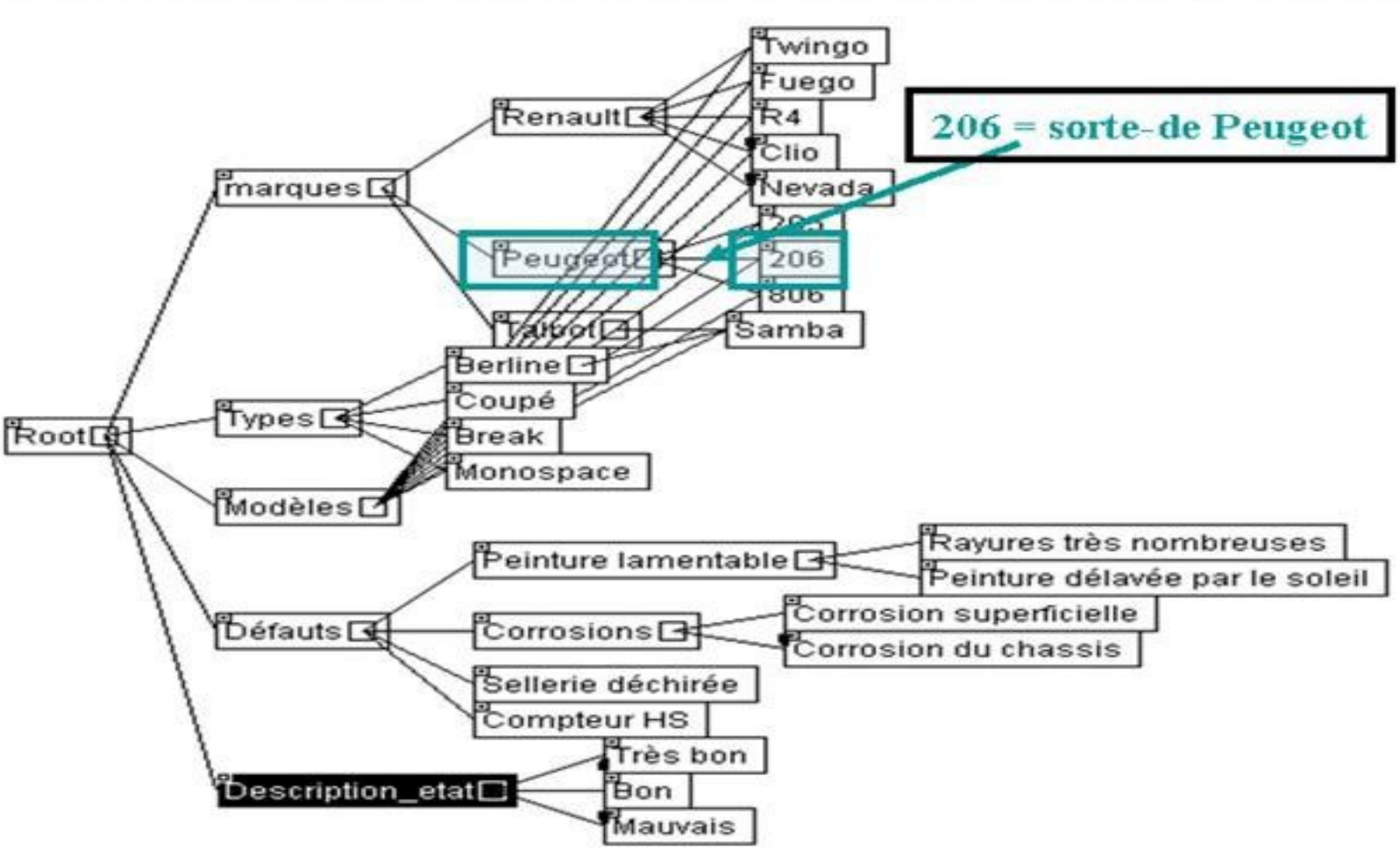
D^c_2 = Condition de vente (détail du plan financier)

Ontologie des attributs de descriptions:

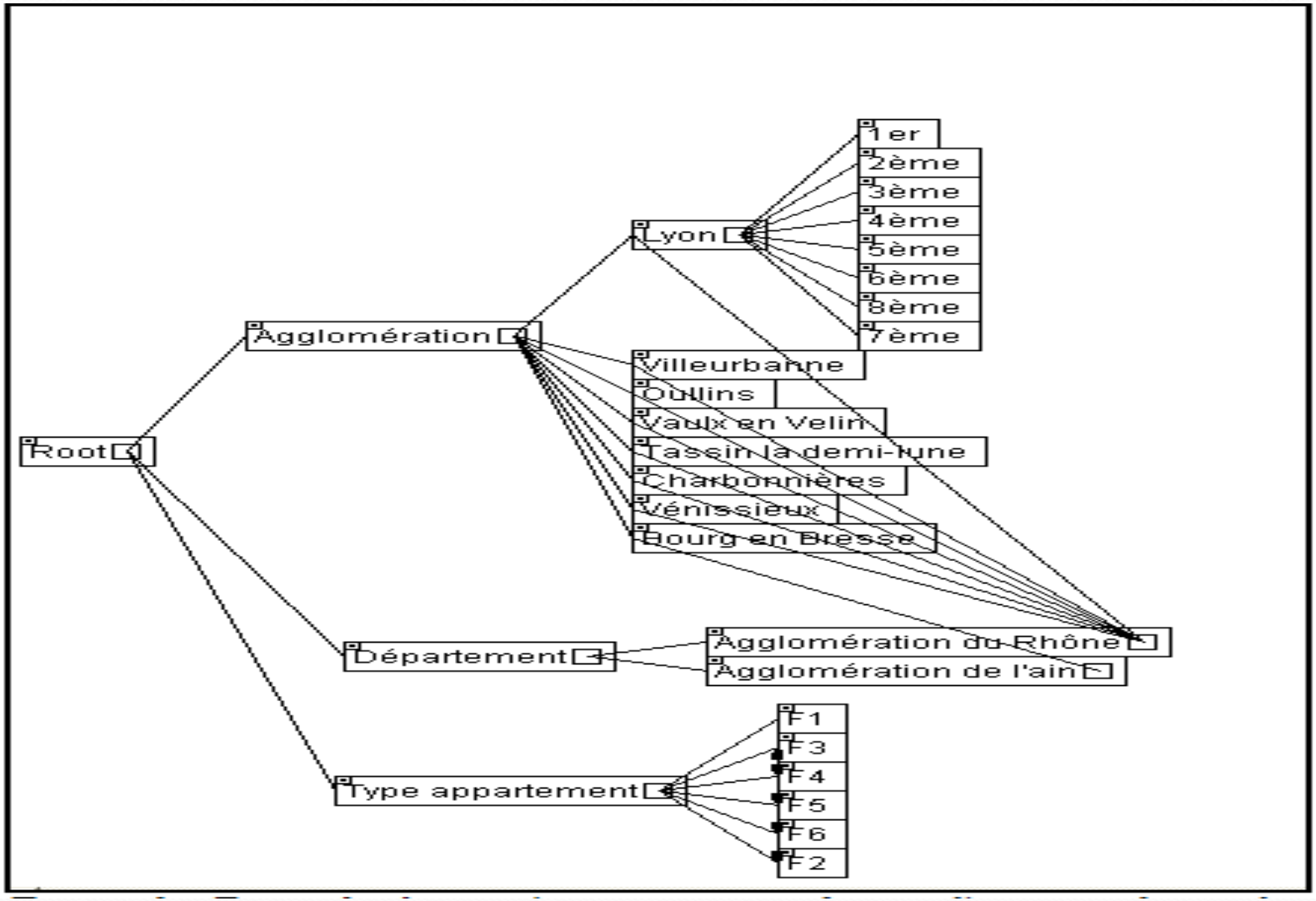
- Une ontologie est un ensemble de termes reliés entre eux par des relations vérifiées.

- En Raisonnement à Partir de Cas, il ne s'agit pas de décrire le monde mais de décrire les relations qui sont vraies entre les termes utilisés pour les valeurs de descripteurs.

Exemple(1):


Root
marques
Types
Modèles
Défauts
Description_etat
Renault
Peugeot
Talbot
Twingo
Fuego
R4
Clio
Nevada
206
806
Samba
Berline
Coupé
Break
Monospace
206 = sorte-de Peugeot
Peinture lamentable
Corrosions
Sellerie déchirée
Compteur HS
Rayures très nombreuses
Peinture délavée par le soleil
Corrosion superficielle
Corrosion du chassis
Très bon
Bon
Mauvais

Exemple(2):


Root
Agglomération
Lyon
1er
2ème
3ème
4ème
5ème
6ème
8ème
7ème
Villeurbanne
Oullins
Vaulx en Velin
Tassin la demi-lune
Charbonnières
Vénissieux
Bourg en Bresse
Département
Agglomération du Rhône
Agglomération de l'ain
Type appartement
F1
F3
F4
F5
F6
F2

Qu'est-ce qu'une base de cas ?

- Une base de cas est une collection de cas de résolution du même problème.
- Les bases de cas peuvent contenir quelques dizaines de cas jusqu'à plusieurs dizaines de milliers.
- Lorsqu'il y a relativement peu de cas, il est important de posséder des connaissances relativement importantes sur le domaine.
- Dans le cas des bases de cas très importantes, des traitements d'apprentissage automatique peuvent être tentés si les cas sont réellement comparables les uns avec les autres..

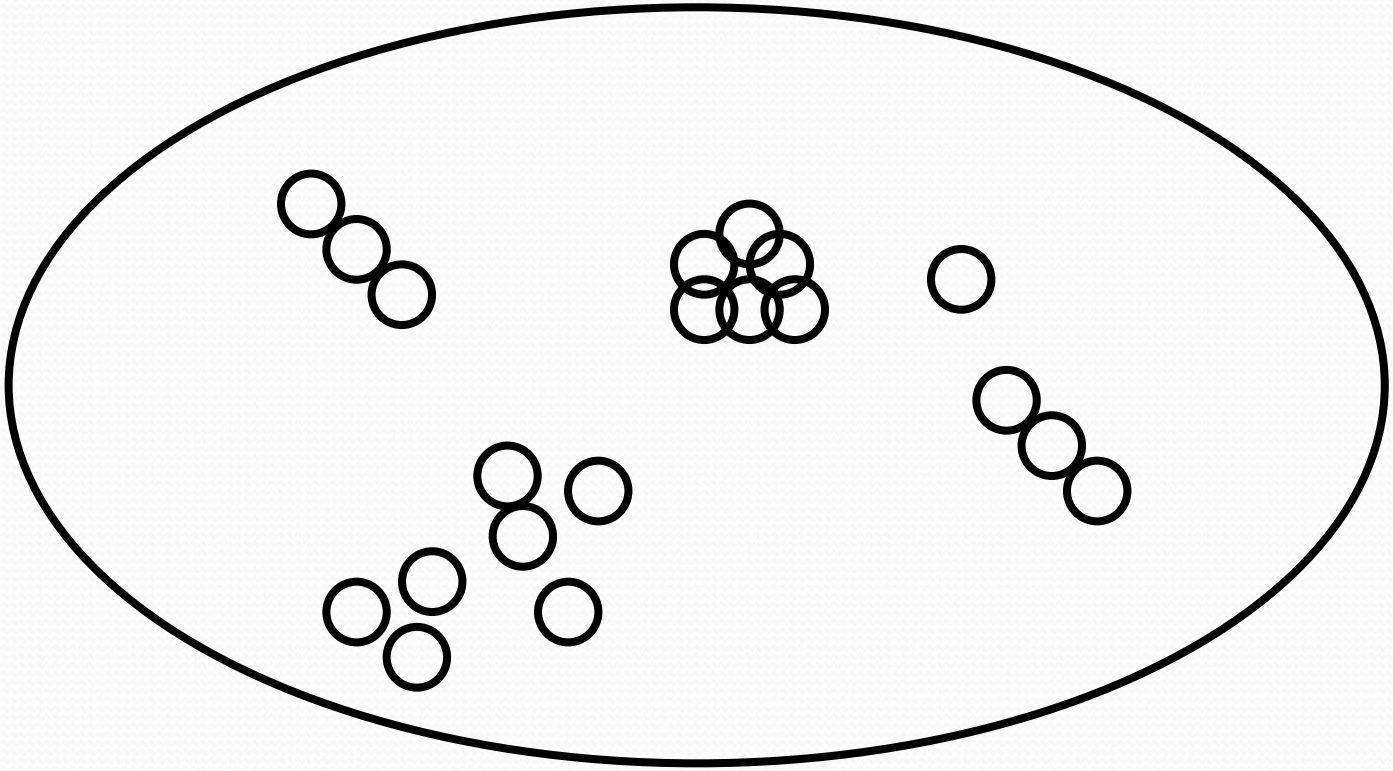
Extrait d'une base de cas:

Descripteurs Solution

Descripteur Problème

Attribut	Type attribut	Cas 1	Cas 2	Cas 3
Surface	Réel	55	35	55
Lieu Départ.	Symbole	Rhône	Rhône	Ain
Lieu Ville	Symbole	Lyon	Lyon	Bourg en Bresse
Type appart.	Symbole	F2	F2	F3
Prix	Réel	20000	45000	15000

Répartition des cas dans une base:



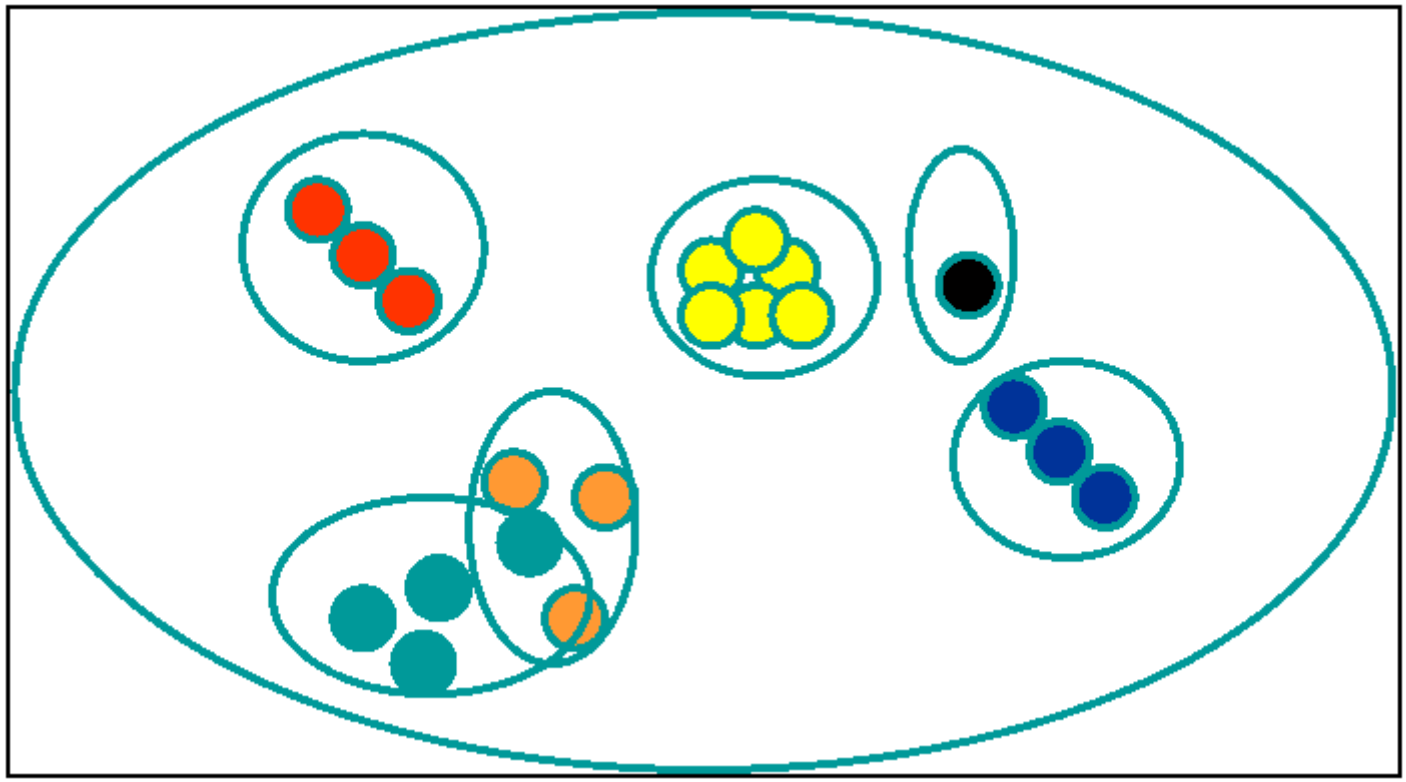
Les cas sont représentés sur le plan des solutions.
Les cas proches ont des solutions proches.

Quel est le principe de résolution : choix d'un cas source dans la base de cas ?

- Dans une base de cas assez grande, il y a peu de chances d’arriver à adapter n'importe quelle solution pour un nouveau problème.

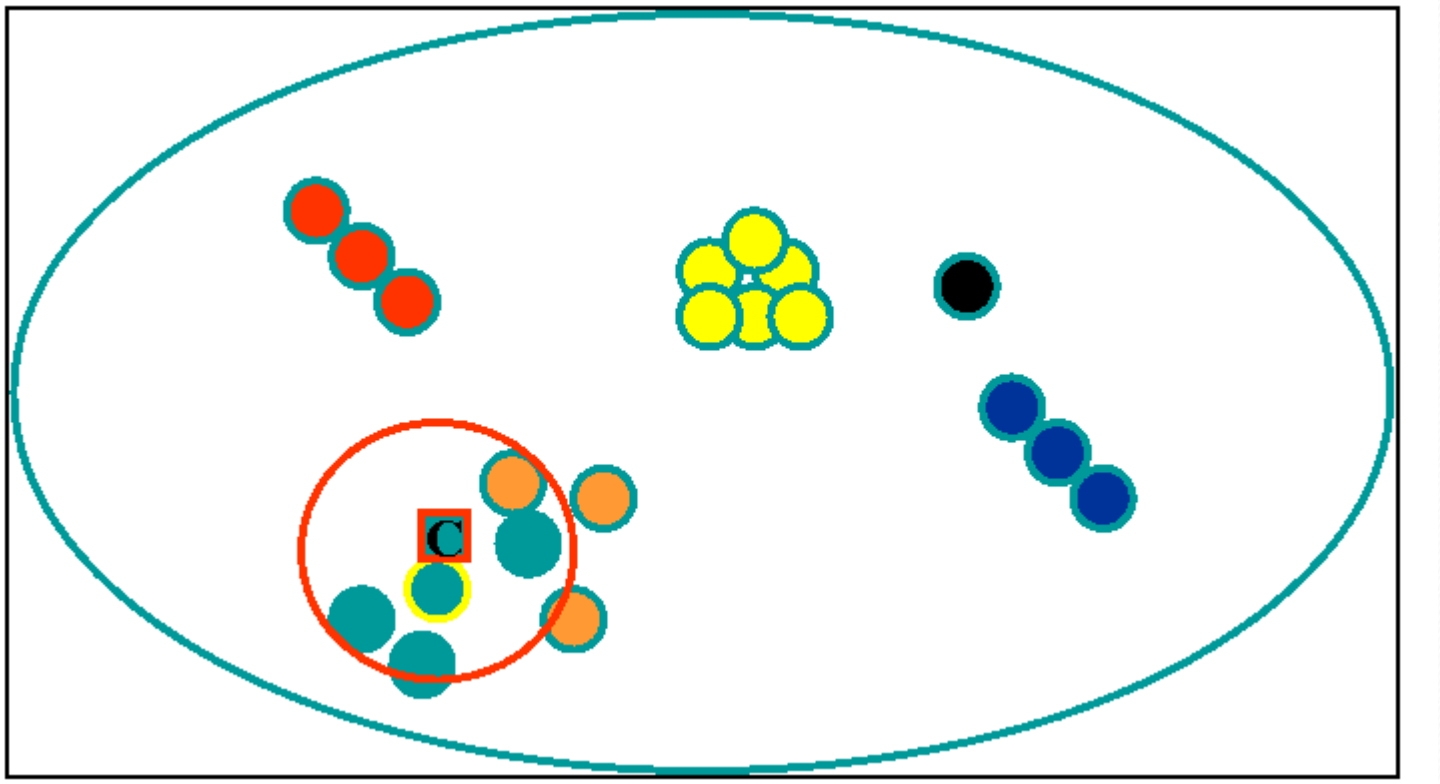
- Il existe un seuil de similarité au-delà duquel il n’est plus possible d’adapter une solution pour une autre et de plus la méthode d'adaptation peut elle-même changer selon la "zone" de solution dans laquelle elle se situe.

Différentes classes de solutions:



Différentes classes de solution correspondant à différentes méthodes d'adaptation avec leurs enveloppes de similarité respectives.

Choix du cas source:

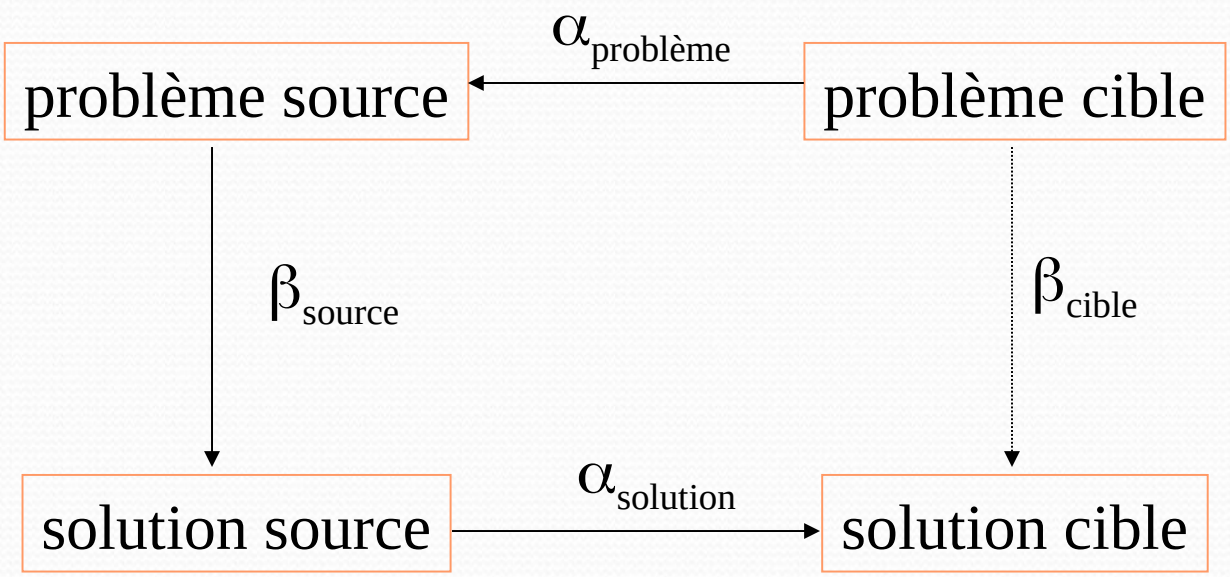


Le nouveau cas cible C est proche de 4 solutions "vertes" et d'une solution "orange". C'est la classe de solution de type "orange" qui est choisie. Le cas "vert" cerclé de "jaune" est le plus proche dans la classe de solution, il est choisi comme cas source pour l'adaptation.

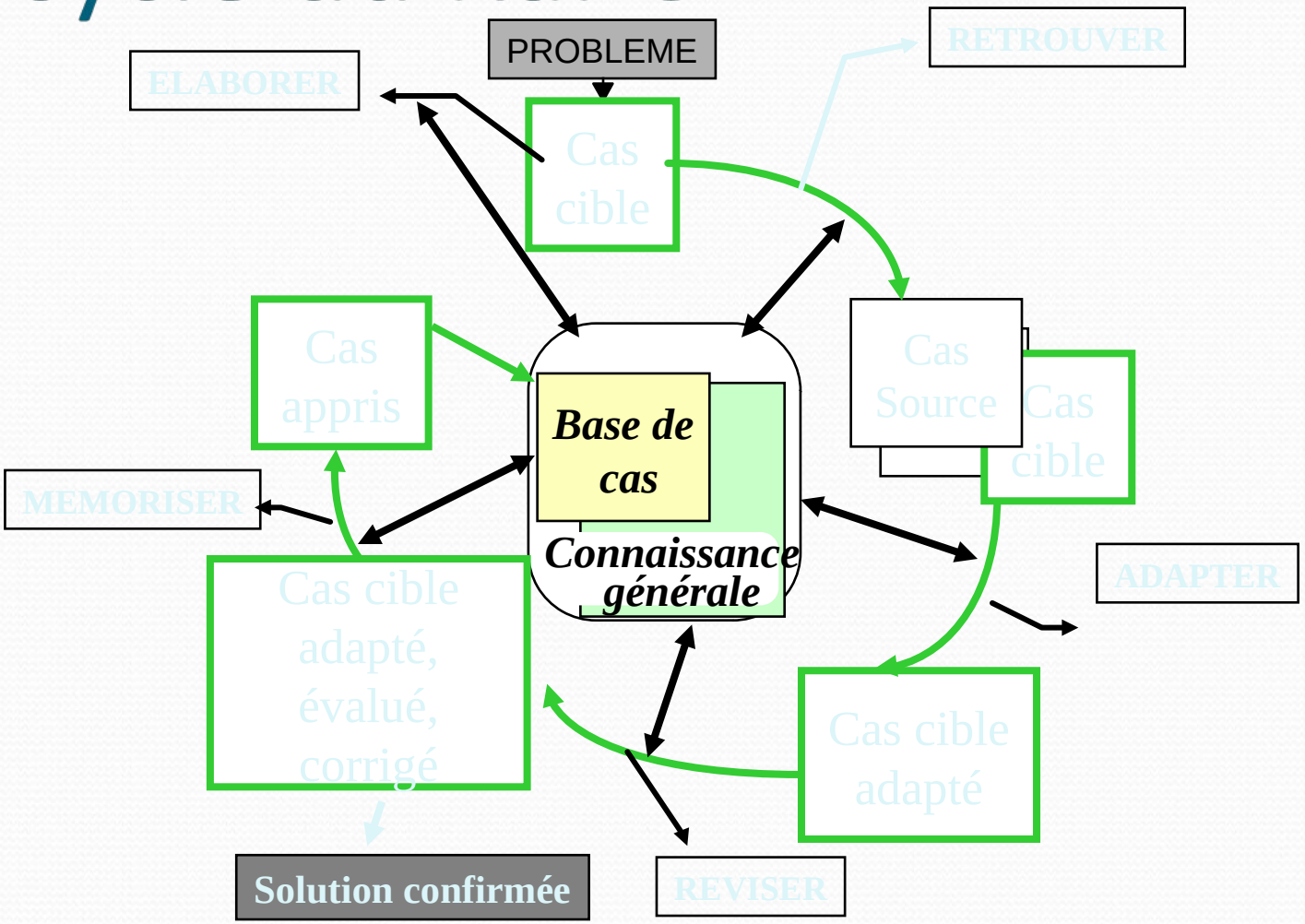
Principes du RàPC:

- Le carré d ’analogie
- Le cycle du RàPC

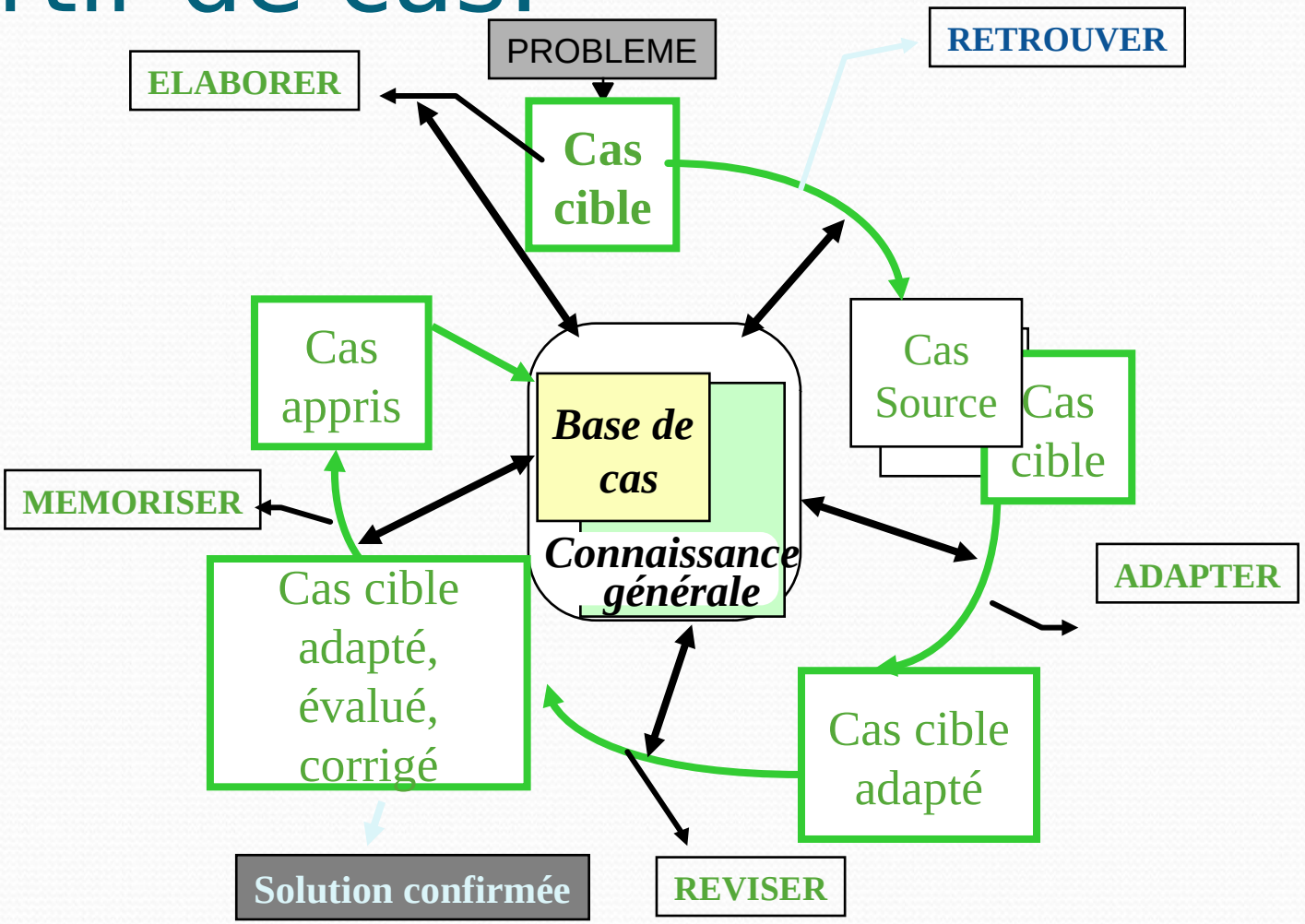
Le carré d'analogie



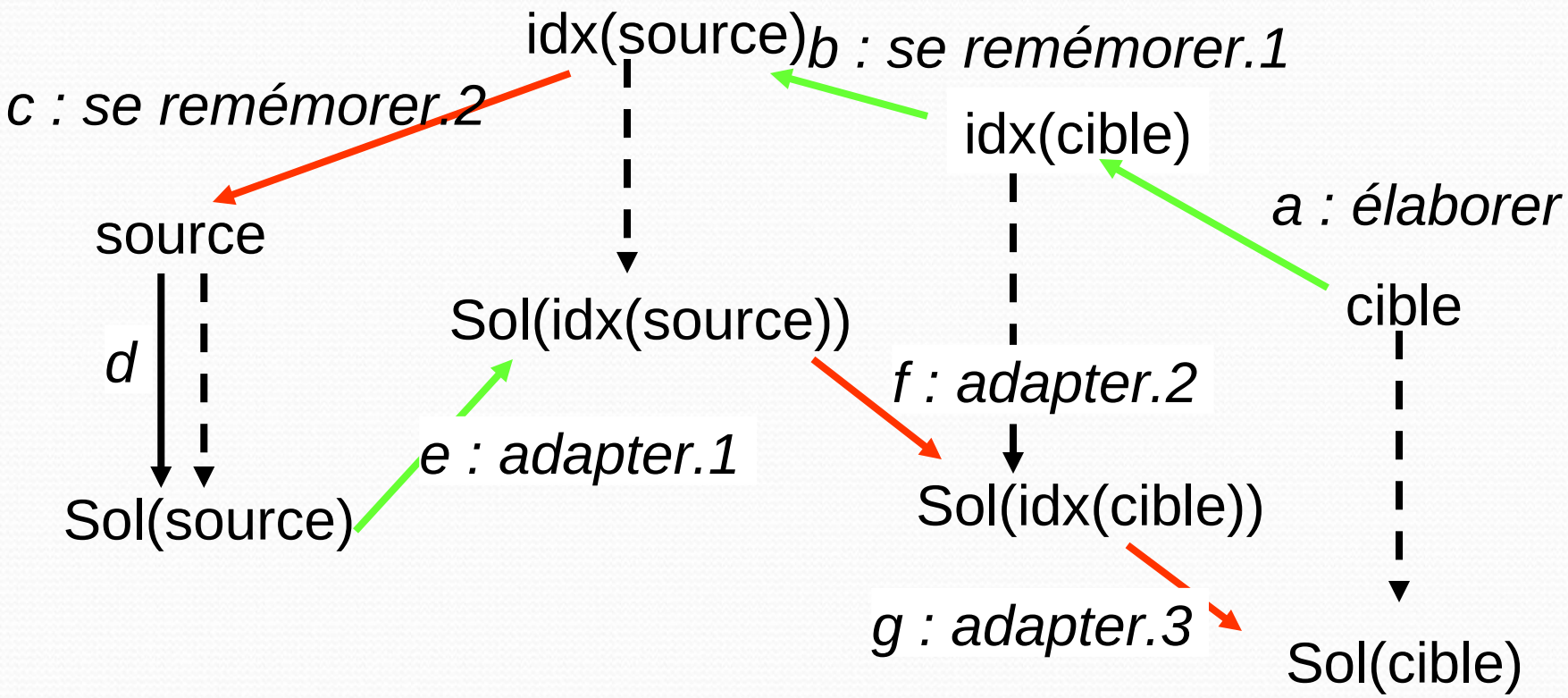
Le cycle du RàPC



Les étapes du raisonnement à partir de cas:



Analogie et cycle revisités



Élaborer:

- Consiste à chercher une solution similaire à partir de l'énoncé d 'un problème.
- Compléter et/ou filtrer la description du problème en se fondant sur les connaissances disponibles sur l'adaptabilité
- Commencer à résoudre le problème

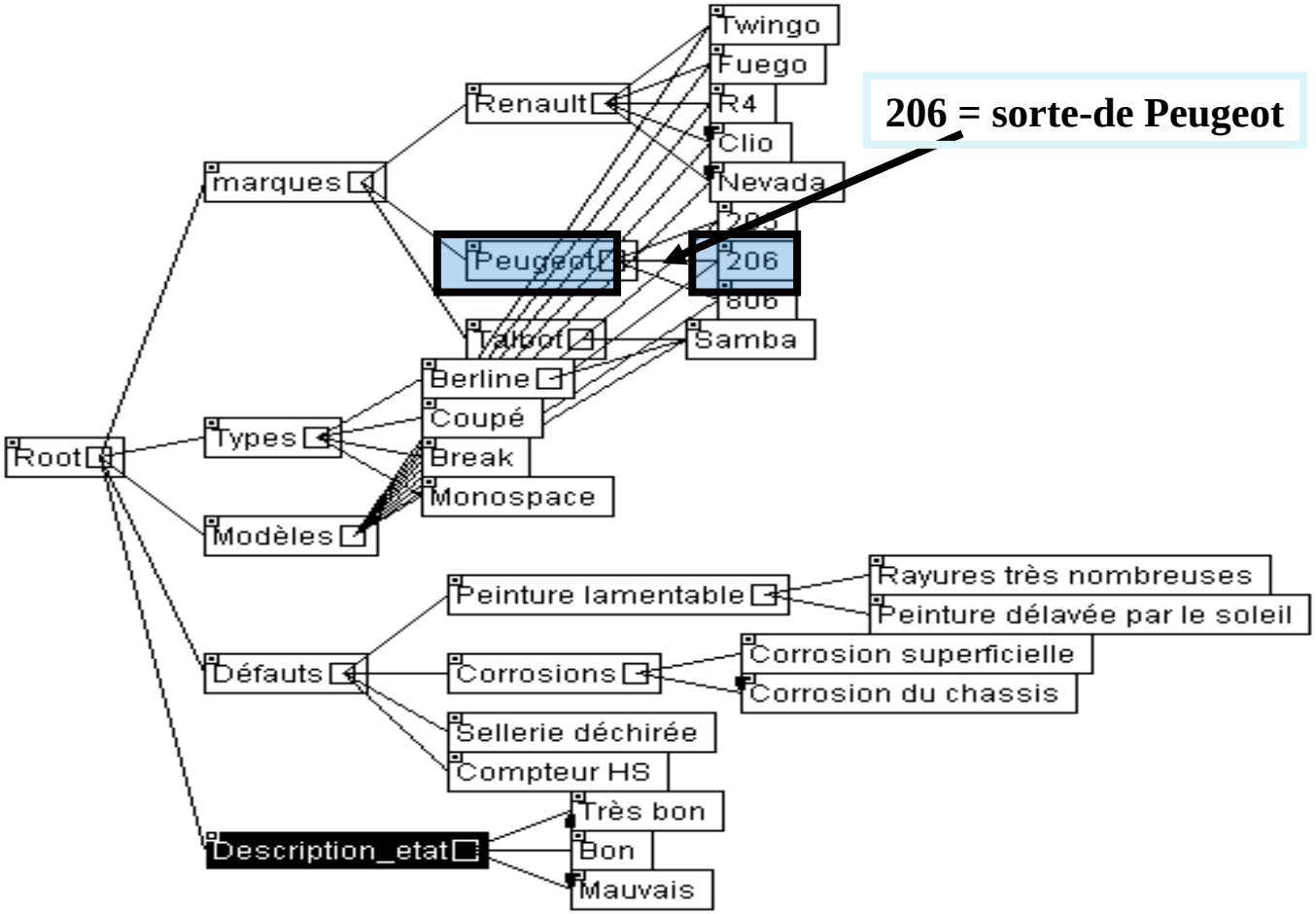
⇒ orienter la recherche d 'une solution adaptable.

Illustration sur la base de cas de vente d'automobiles d'occasion(1):

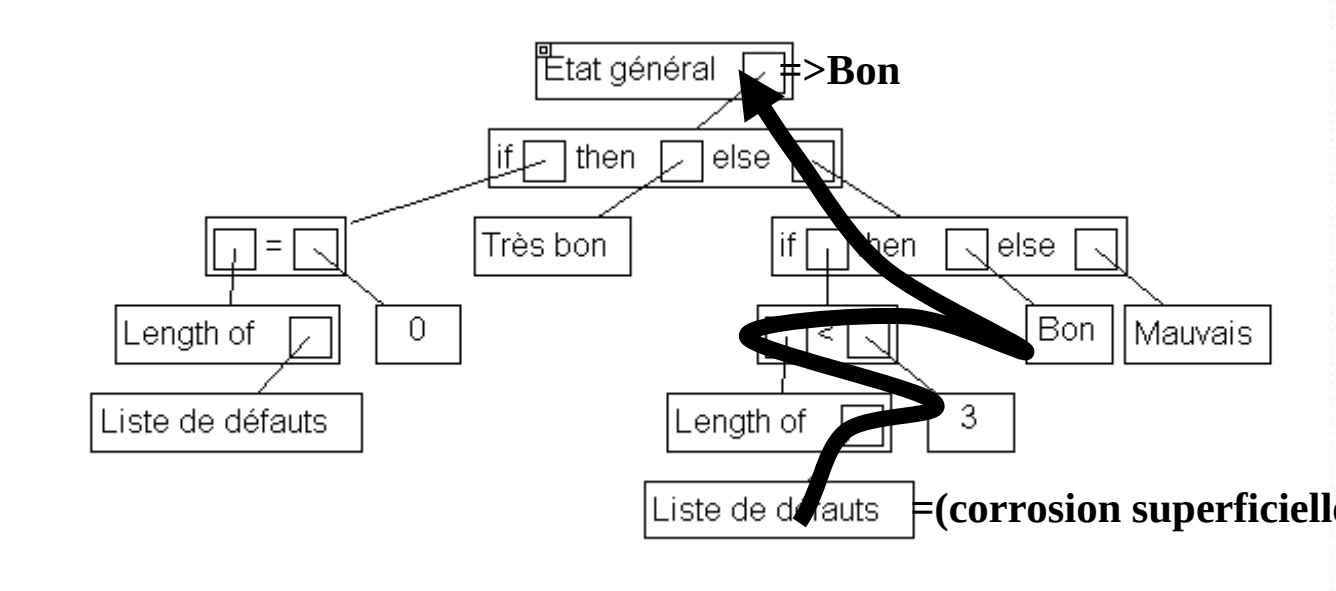
- Cas cible non élaboré:

Nom de l'attribut	Type de l'attribut	Exemple de valeur du de l'attribut
Etat Général	Symbole (déduit)	??
Kilométrage	Réel	198000
Age du véhicule	Réel	10
Marque	Symbole (déduit)	??
Modèle	Symbole	206
Type de véhicule	Symbole	Coupé
Liste de défauts constatés	Liste de Symboles	(Corrosion superficielle)
tPrix de vente (solution)	Réel	???

Élaboration / Ontologie du domaine de la vente:



Règle d’élaboration:



Règle d'élaboration de l'état général d'un véhicule à partir de la connaissance de la liste des défauts décrits

Illustration sur la base de cas de vente d'automobiles d'occasion(2):

- Cas cible élaboré:

Nom de l'attribut	Type de l'attribut	Exemple de valeur du de l'attribut
Etat Général	Symbole (déduit)	Bon
Kilométrage	Réel	198000
Age du véhicule	Réel	10
Marque	Symbole (déduit)	Peugeot
Modèle	Symbole	205
Type de véhicule	Symbole	Coupé
Liste de défauts constatés	Liste de Symboles	(Corrosion superficielle)
Prix de vente (solution)	Réel	???

Chercher:

- Similarité = degré d’appariement entre deux cas :
  - Recherche des correspondances entre descripteurs.
  - Calcul du degré d’appariement des descripteurs.
  - Pondération éventuelle des descripteurs dans le cas.

Chercher: exemple

Nom de l'attribut	Type de l'attribut	Poids de l'influence de l'attribut sur la solution
Etat Général	Symbole (déduit)	20%
Kilométrage	Réel	35%
Age du véhicule	Réel	25%
Marque	Symbole (déduit)	5%
Modèle	Symbole	5%
Type de véhicule	Symbole	10%
Liste de défauts constatés	Liste de Symboles	Pas pris en compte
Prix de vente (solution)	Réel	???

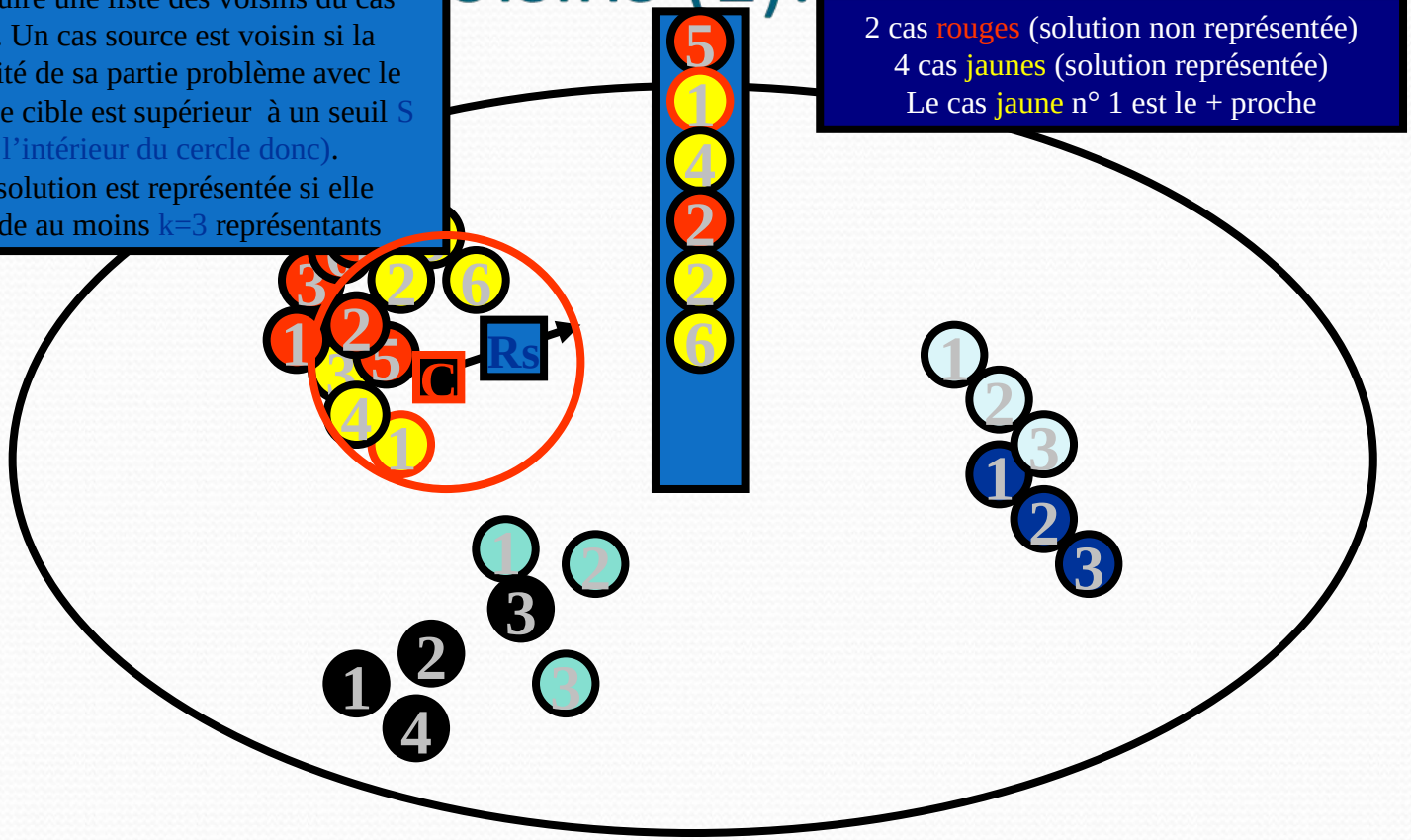
Poids des différents attributs = importance de l' influence relative de ces attributs sur la solution.

Algorithme KPPV

K Plus Proches Voisins (1):

Construire une liste des voisins du cas cible. Un cas source est voisin si la similarité de sa partie problème avec le problème cible est supérieur à un seuil S (à l'intérieur du cercle donc).
Une solution est représentée si elle possède au moins k=3 représentants

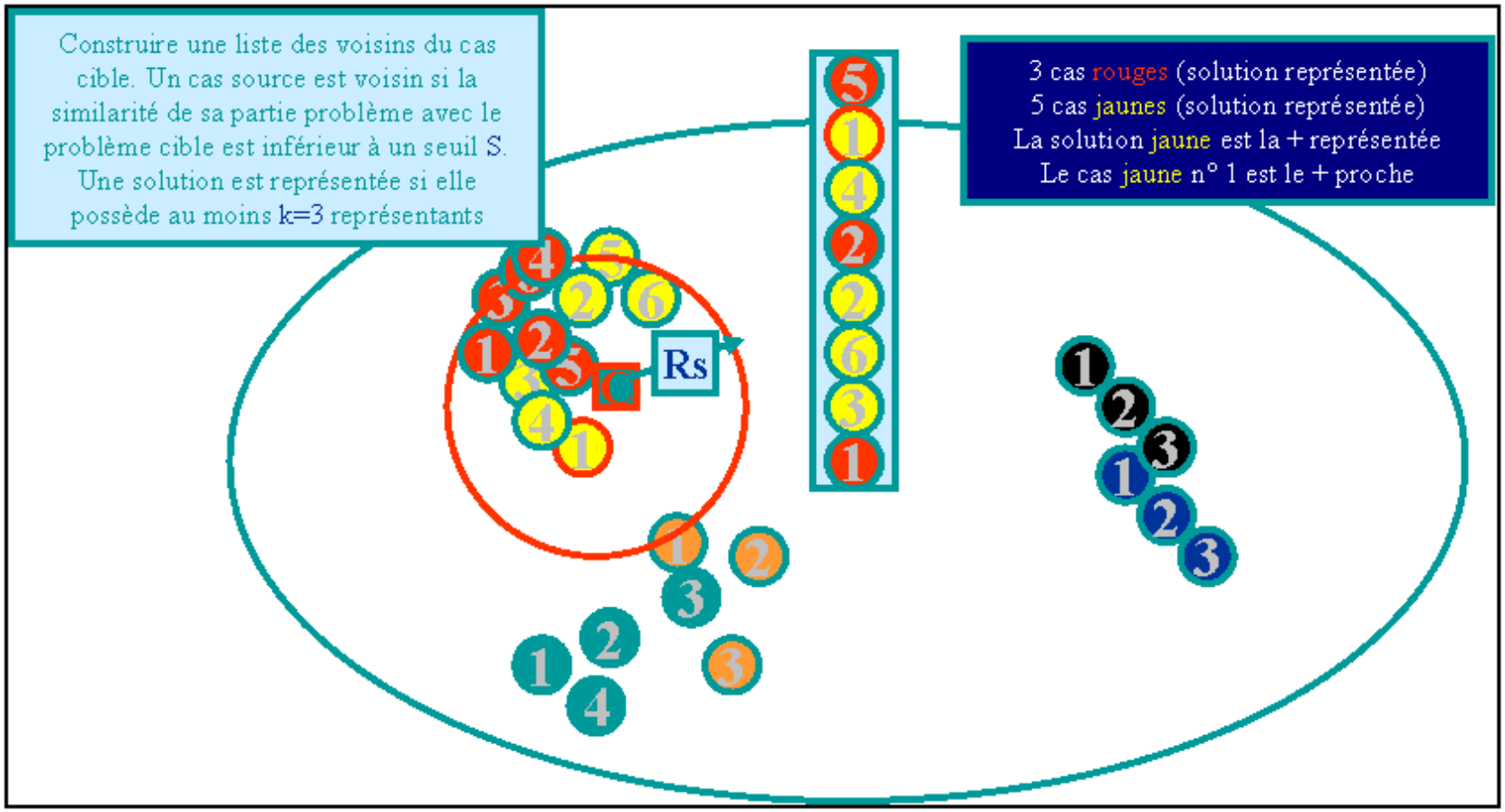
2 cas rouges (solution non représentée)
4 cas jaunes (solution représentée)
Le cas jaune n° 1 est le + proche



Scénario de déroulement de l'algorithme KPPV sélectionnant le cas 1 (de classe solution "jaune") pour le cas cible C.
Seule la solution jaune était éligible (représentation > 3)

Algorithme KPPV

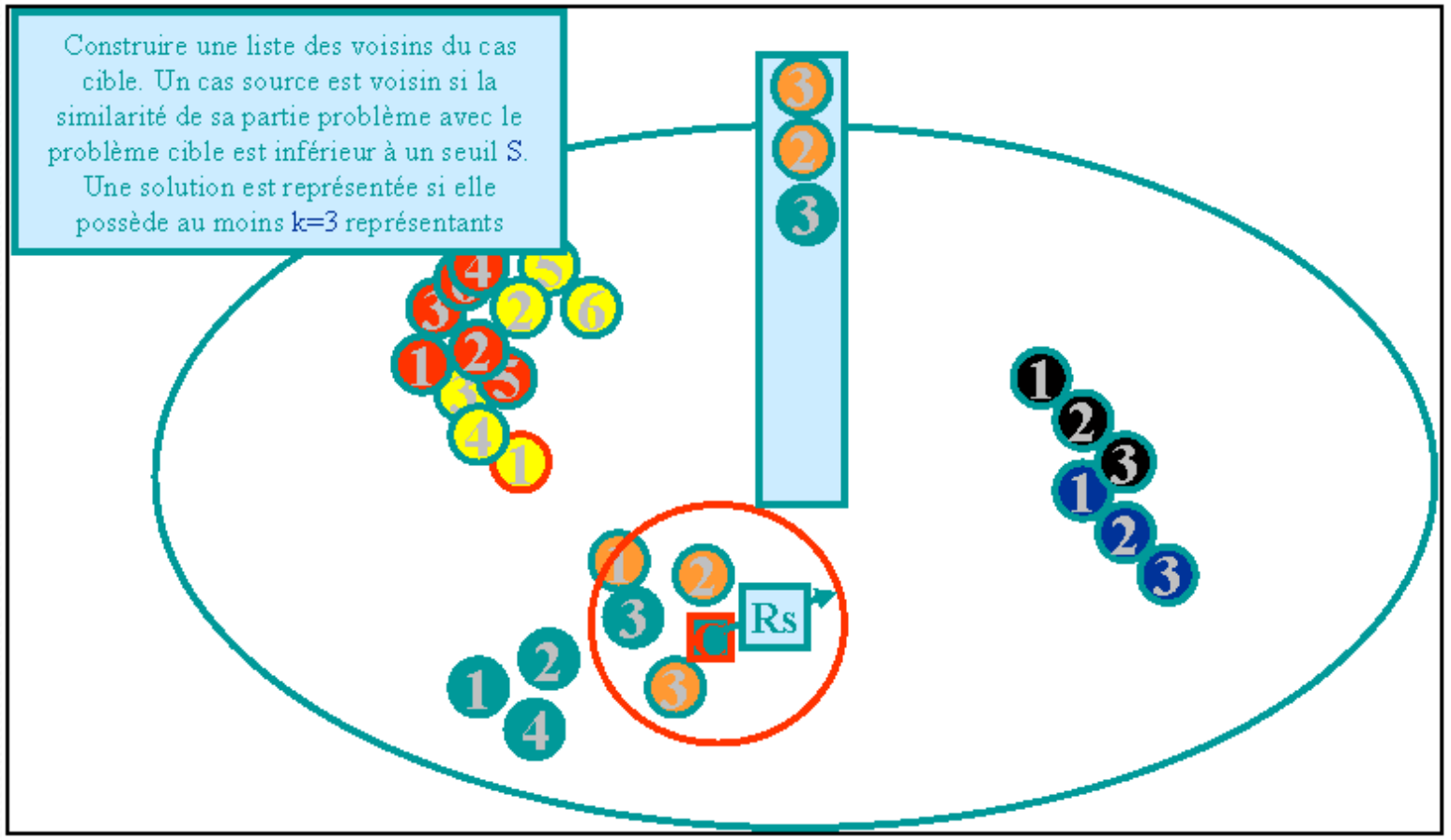
K Plus Proches Voisins (2):



un seuil plus bas (Rs plus grand) la solution "rouge" devient éligible. Elle n'est pas la plus représentée. Le cas 1 (jaune) est encore sélectio

Algorithme KPPV

K Plus Proches Voisins (3):

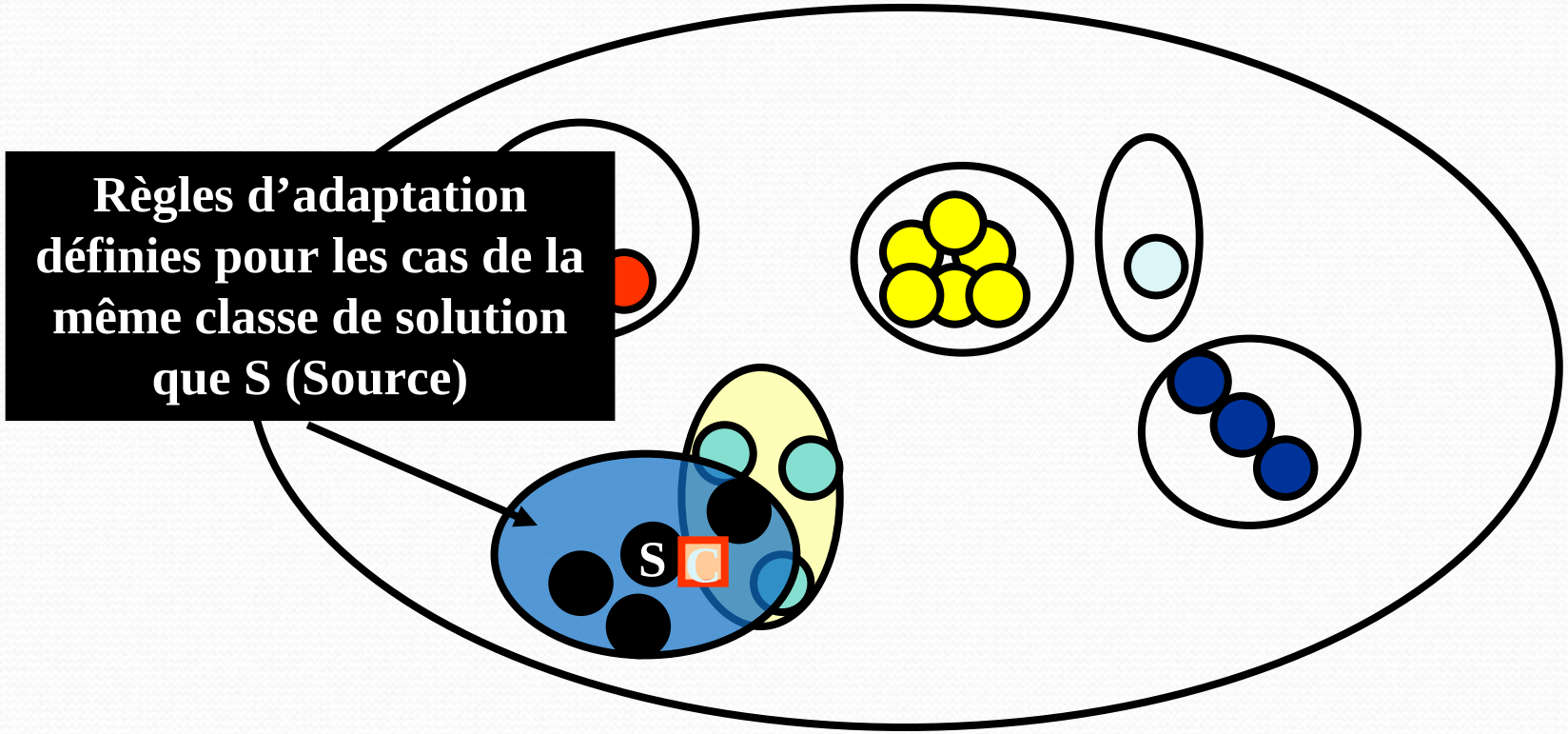


Dans ce scénario, aucune solution n'est éligible. Aucune expérience ne peut être rationnellement remémorée. Ce cas cible devra être résolu par une autre méthode.

Adapter : la problématique

- Il s’agit de réutiliser la solution d’un cas proche,
- en supposant qu’il est possible d’adapter ce cas,
- Il est plus facile de l’adapter que d’essayer de le résoudre directement..

Zone d’appartenance à une classe:



Les règles d'adaptation de la solution s'expriment en fonction des écarts relevés entre les description des problèmes cibles et source.

Exemple : Connaissance / Similarité

Nearest Neighbor

IMPORTANCE LEVEL:

Exact Ignore

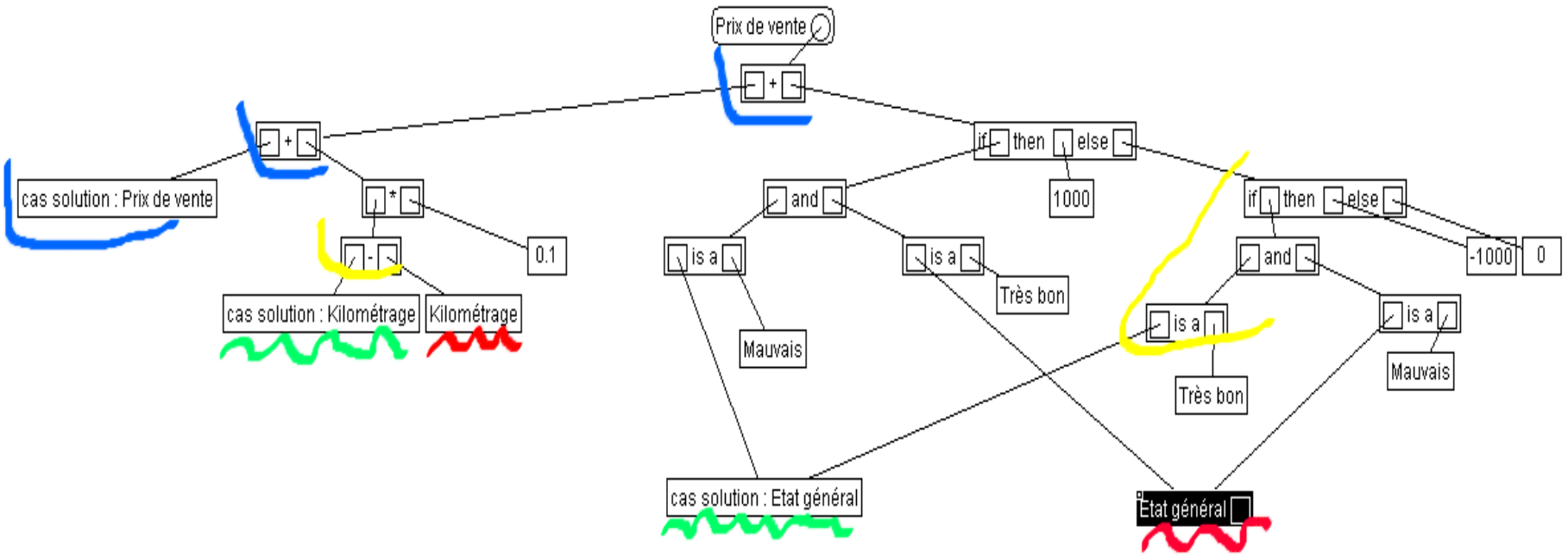
1 2 4 8 16

Global Weight Vector : Default Weight Vec

Field Name	Field Value	Field Type
Age du véhicule		Real
cas solution		Case
Etat général	2 (17%)	Symbol (Formula)
Kilométrage	8 (67%)	Integer
Liste de défauts		List of Symbol
Marque	Exact Match	Symbol (Formula)
Modèle	2 (17%)	Symbol
Prix de vente		Real
Type de véhicule		Symbol

Le descripteur solution "Prix de vente" est influencé par des écarts sur l'état général, le kilométrage et le modèle. Il y aura des rgèles différentes pour chaque marque de véhicule.

Exemple: connaissance/adaptation



A l'issue de l'adaptation, une solution hypothétique est proposée. Sa valeur n'est pas garantie.

Une originalité importante du RàPC est que le cycle de raisonnement prévoit explicitement cette révision.

Évaluer/Réviser

- L ’objectif est de faire le bilan d ’un cas avant sa mémorisation / apprentissage :
- Vérification par introspection dans la base de cas.
- Utilisation d’un système de vérification (contrôle de cohérence globale, simulateur, etc.).
- Retour du « monde réel ».
  - ⇨intégration des révisions dans le cas

Mémoriser : vers l'apprentissage

- Ajouter le cas dans la base (selon la qualité des cas par exemple).
- Organiser le cas dans la base : l'insérer dans un réseau d 'explications.
- Indexer le cas dans la base.
- Synthétiser des connaissances nouvelles.

Conclusion:

- Le cadre proposé ne doit pas être vu comme un modèle formel , mais comme un cadre de modélisation souple.

- Ceci fournit au concepteur une première description des fonctionnalités du RàPC et constitue donc un guide pour concevoir un nouveau modèle de RàPC.