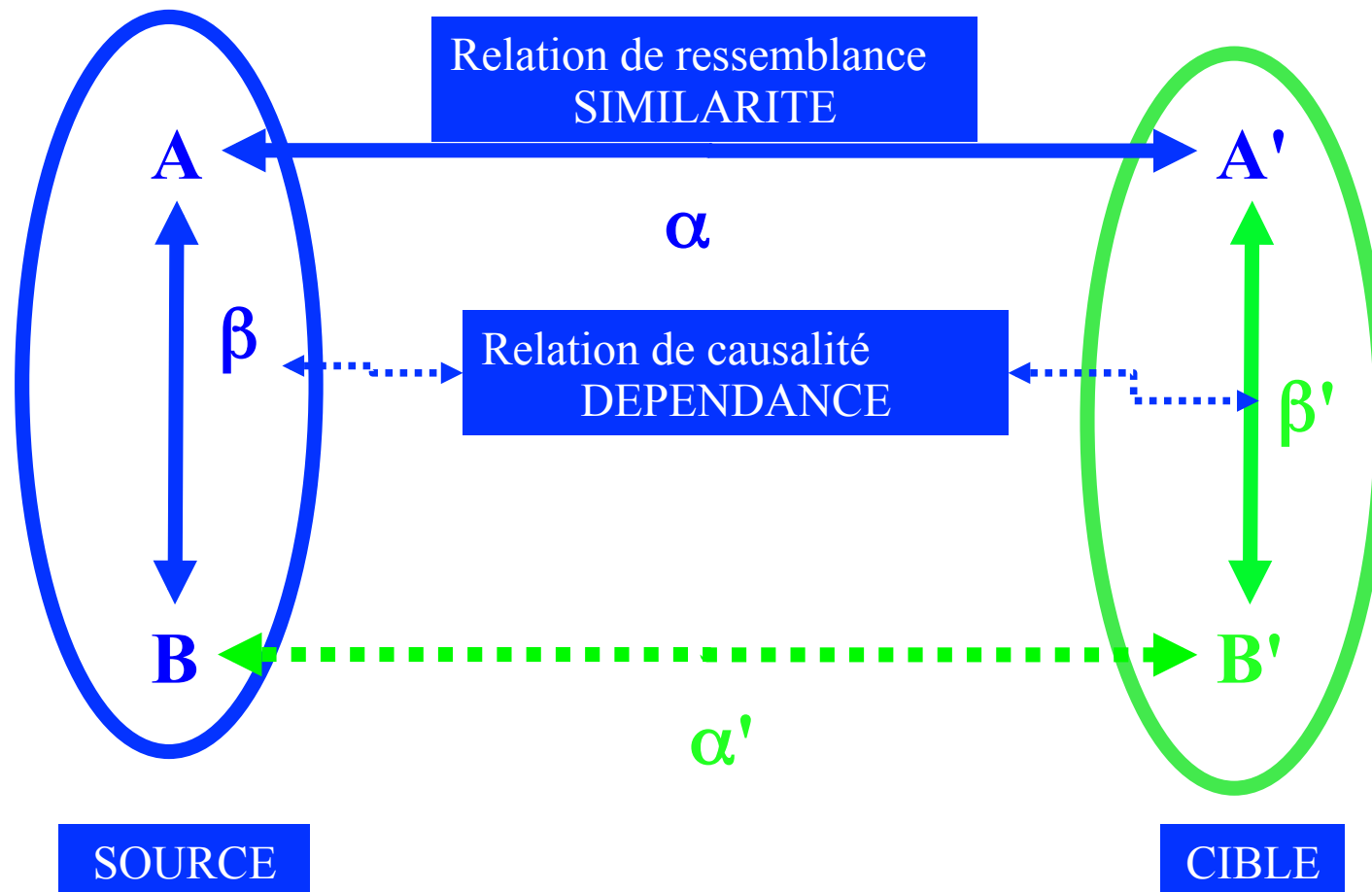


“Du particulier au particulier”

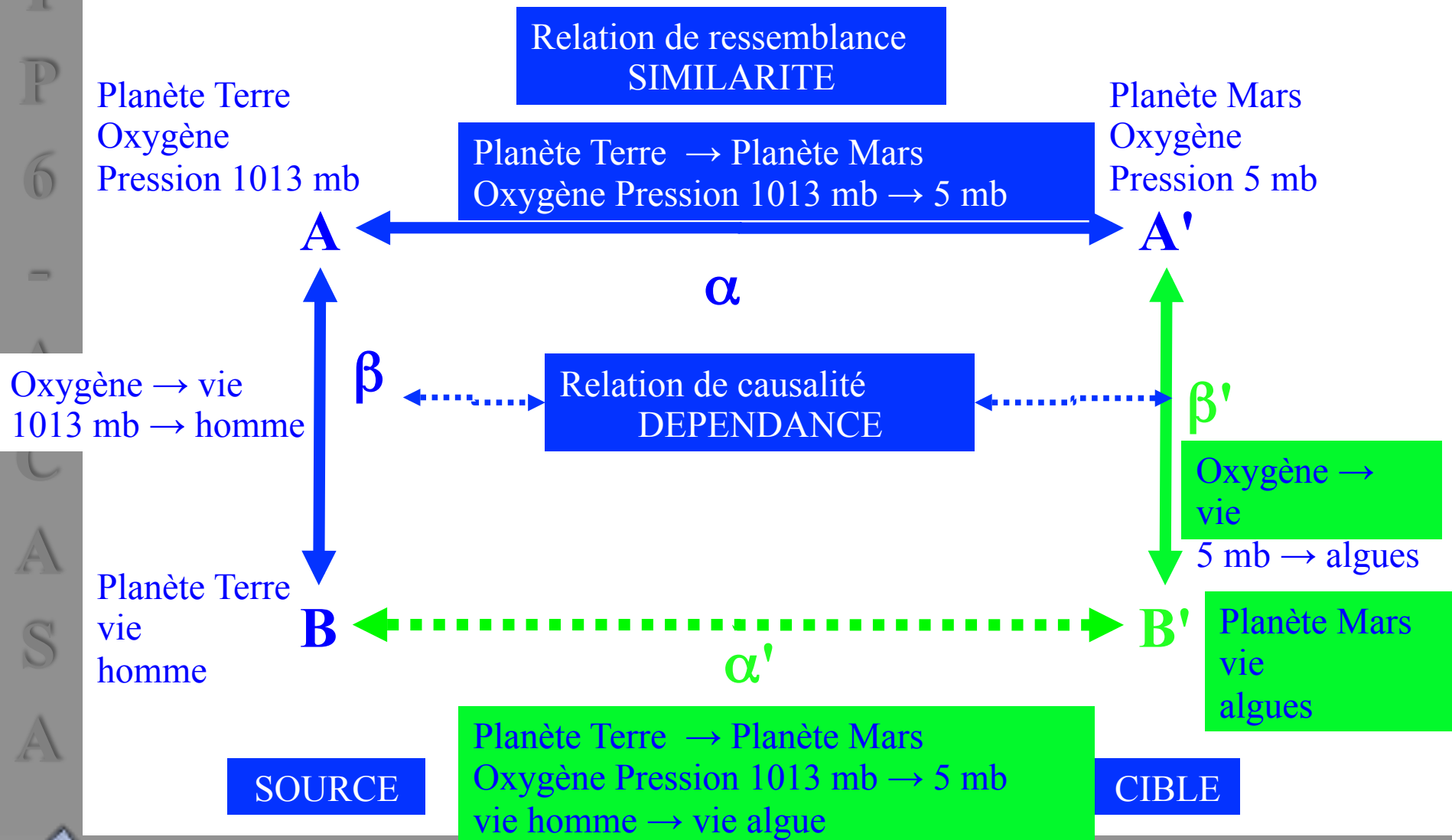
Analogie,
Apprentissage paresseux,
Apprentissage à partir
d'instances
& Raisonnement à partir de cas



Analogie: une approche de Intelligence artificielle



Un exemple d'analogie

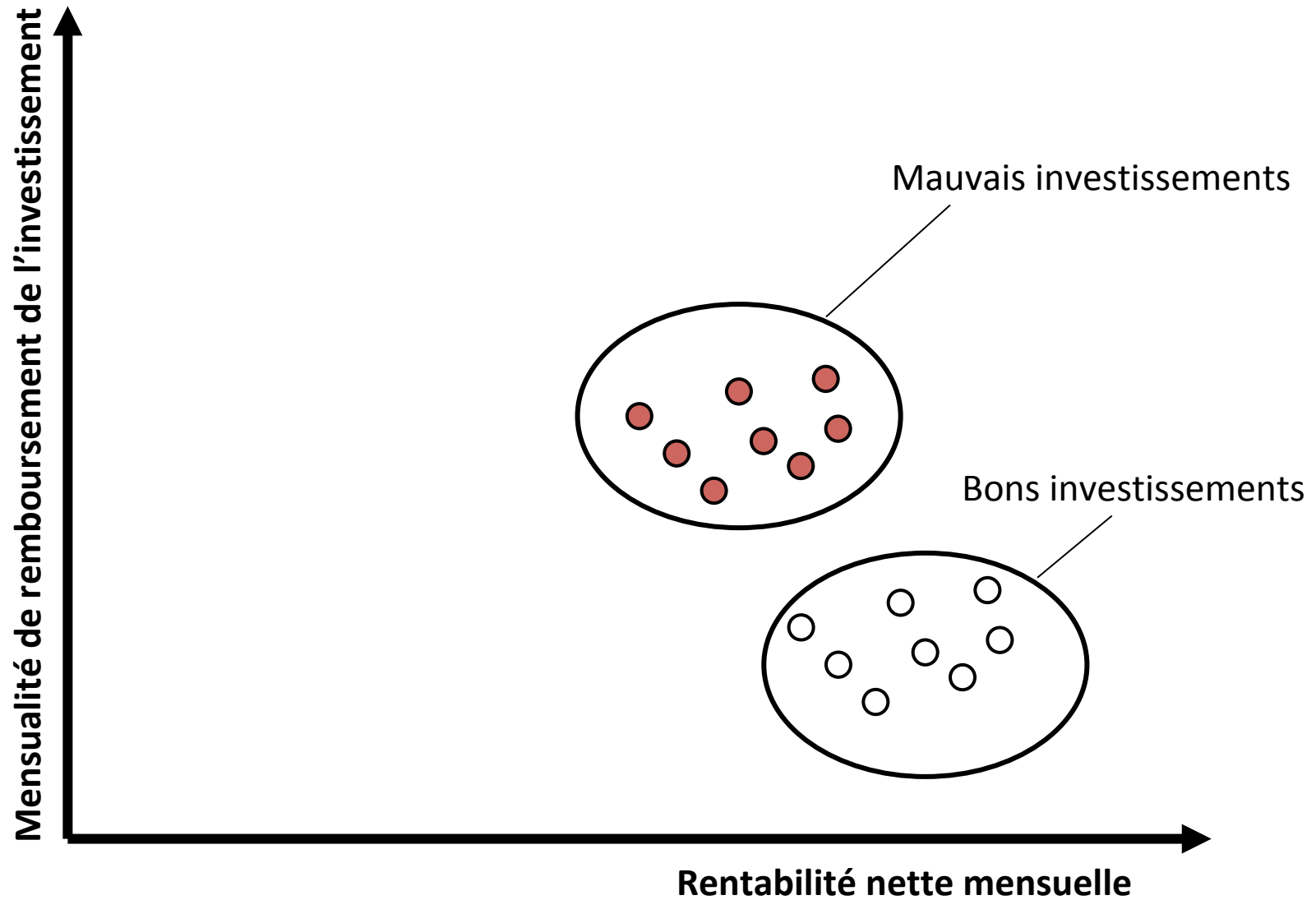


Induction & analogie

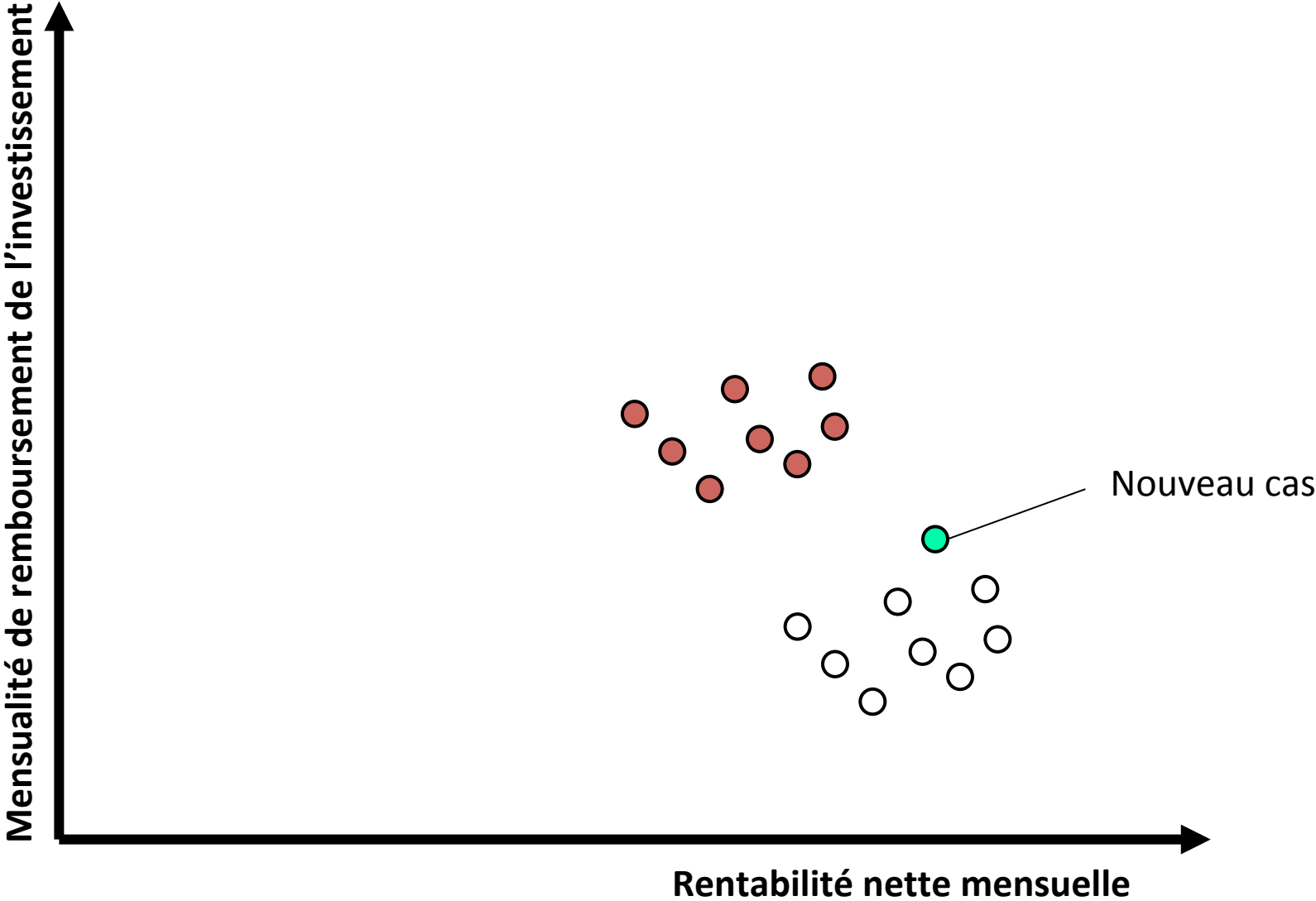
« *L'induction* conclut du particulier à l'universel (*a particulari ad universale*) selon le principe de *généralisation* : ce qui convient à plusieurs choses convient aussi aux autres choses du même genre. — *L'analogie* conclut de la ressemblance *particulière* entre deux choses à la ressemblance *totale*, selon le principe de *spécification* : des choses d'un genre entre lesquelles nous connaissons beaucoup de points d'accord, s'accordent aussi sur les autres points que nous connaissons chez quelques individus de ce genre, mais que nous ne percevons pas chez les autres. — L'induction étend le donné empirique du particulier au général relativement à *plusieurs objets* ; l'analogie étend les *propriétés données* d'une chose à un plus grand nombre de propriétés de *cette même chose*. — *Un en plusieurs*, donc en tous : *induction*; *plusieurs en un* (qui sont généralement en un autre) donc également le reste dans le même: *analogie*. » Logique, E. Kant, p. 144.



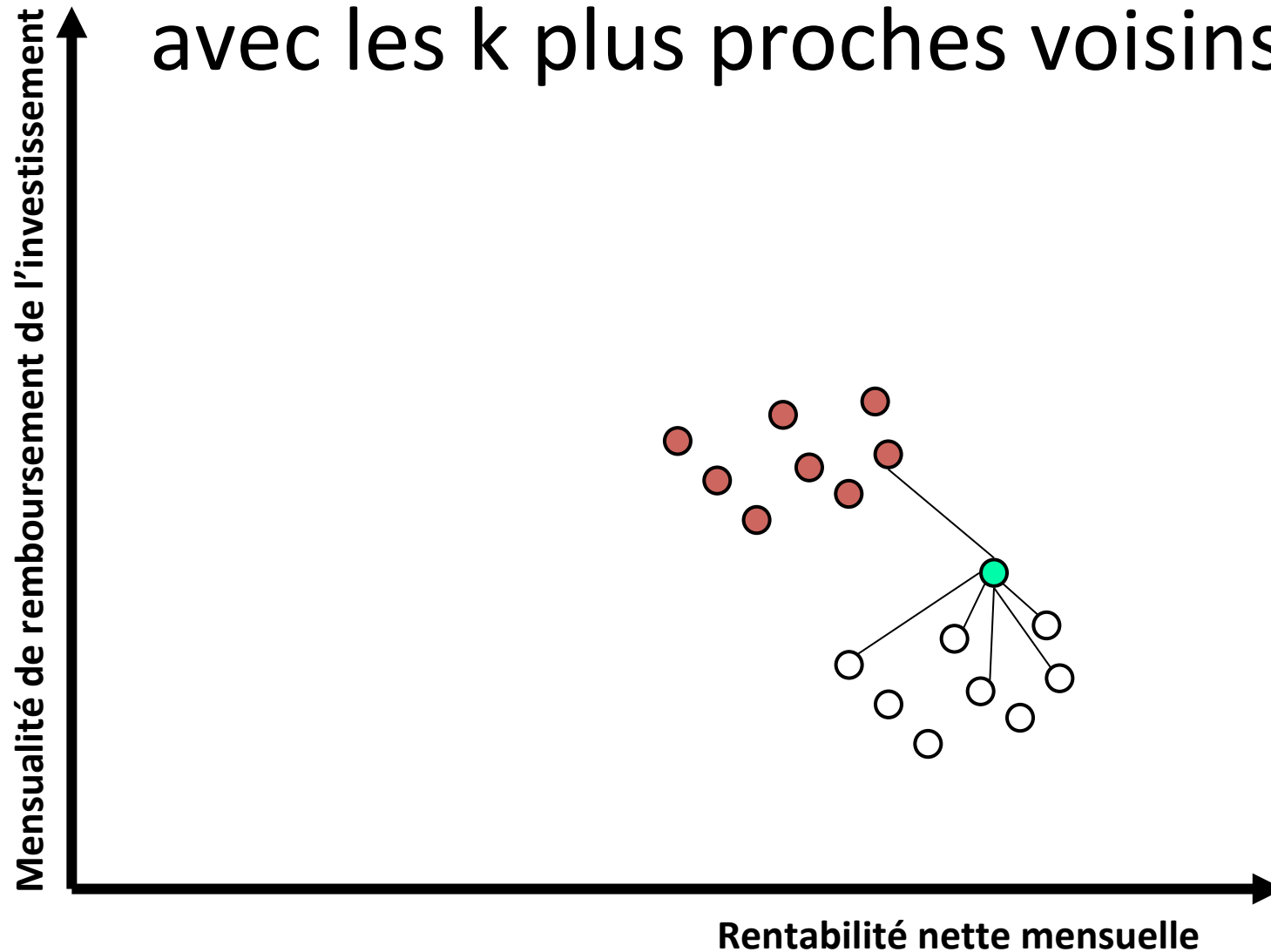
Généralisation – induction



Raisonnement



Calcul des distances avec les k plus proches voisins



Instance-based learning – Lazy learning k plus proches voisins

- Chaque exemple x est représenté par un n -uplet du type: $\langle a_1(x), a_2(x), \dots, a_n(x) \rangle$
- Distance entre deux voisins x_i et x_j :

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{r=n} [a_r(x_i) - a_r(x_j)]^2}$$



Instance-based learning

Algorithme des k plus proches voisins:

- Chaque **exemple** $\langle x, f(x) \rangle$ doit être ajoutée à la liste des **exemples d'apprentissage A**
- Pour classer l'**exemple** x_q , c'est-à-dire pour trouver $f(x_q)$, trouver les k exemples d'apprentissage, c'est-à-dire les k éléments de A $x_{s(1)}, \dots, x_{s(k)}$, les plus proches de x_q
- L'**estimation g de f** est donnée par l'algorithme des k plus proches voisins:

$$g(x_q) = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{i=k} \delta(v, f(x_{s(i)}))$$
$$\delta(a, b) = \begin{cases} 1 & \text{si } a = b \\ 0 & \text{si } a \neq b \end{cases}$$



Exemple: base iris

150 exemples – 5 champs

iris.key

goalfield:5

5.1,3.5,1.4,0.2,Iris-setosa

4.9,3.0,1.4,0.2,Iris-setosa

4.7,3.2,1.3,0.2,Iris-setosa

4.6,3.1,1.5,0.2,Iris-setosa

5.0,3.6,1.4,0.2,Iris-setosa

5.4,3.9,1.7,0.4,Iris-setosa

4.6,3.4,1.4,0.3,Iris-setosa

5.0,3.4,1.5,0.2,Iris-setosa

4.4,2.9,1.4,0.2,Iris-setosa

4.9,3.1,1.5,0.1,Iris-setosa

5.4,3.7,1.5,0.2,Iris-setosa

4.8,3.4,1.6,0.2,Iris-setosa

4.8,3.0,1.4,0.1,Iris-setosa

7.0,3.2,4.7,1.4,Iris-versicolor

6.4,3.2,4.5,1.5,Iris-versicolor

6.9,3.1,4.9,1.5,Iris-versicolor

5.5,2.3,4.0,1.3,Iris-versicolor

6.5,2.8,4.6,1.5,Iris-versicolor

5.7,2.8,4.5,1.3,Iris-versicolor

6.3,3.3,4.7,1.6,Iris-versicolor

4.9,2.4,3.3,1.0,Iris-versicolor

6.6,2.9,4.6,1.3,Iris-versicolor

5.2,2.7,3.9,1.4,Iris-versicolor

5.0,2.0,3.5,1.0,Iris-versicolor

5.9,3.0,4.2,1.5,Iris-versicolor

6.0,2.2,4.0,1.0,Iris-versicolor



Examples

Case-based Reasoning Shell

AIAI CBR AIAI Controls AIAI Optimiser

Current Case: 69 Class: Iris-versicolor (Casebase of 150)
Nearest Neighbours:
Case: 120 Score: 3.86 Class: Iris-virginica
Case: 88 Score: 3.78 Class: Iris-versicolor
Case: 73 Score: 3.78 Class: Iris-versicolor
Case: 79 Score: 3.72 Class: Iris-versicolor
Case: 55 Score: 3.71 Class: Iris-versicolor
Analysis:
Class: Iris-virginica Count: 1
Class: Iris-versicolor Count: 4
Decision: Iris-versicolor
Accuracy: 69/69 (100.0%)

Step Run Stop Reset

Load

Case-based Reasoning Shell

CBR AIAI Controls AIAI Optimiser

Current Case: 71 Class: Iris-versicolor (Casebase of 150)
Nearest Neighbours:
Case: 139 Score: 3.92 Class: Iris-virginica
Case: 128 Score: 3.88 Class: Iris-virginica
Case: 150 Score: 3.87 Class: Iris-virginica
Case: 127 Score: 3.82 Class: Iris-virginica
Case: 57 Score: 3.77 Class: Iris-versicolor
Analysis:
Class: Iris-virginica Count: 4
Class: Iris-versicolor Count: 1
Decision: Iris-virginica
Accuracy: 70/71 (98.59%)

Step Run Stop Reset

Load

Raffinement de l'algorithme

- On ajoute **un poids w_i** qui avantage les voisins les plus proches

- Ainsi, l'estimation g de la classe f se fait de la

façon suivante:

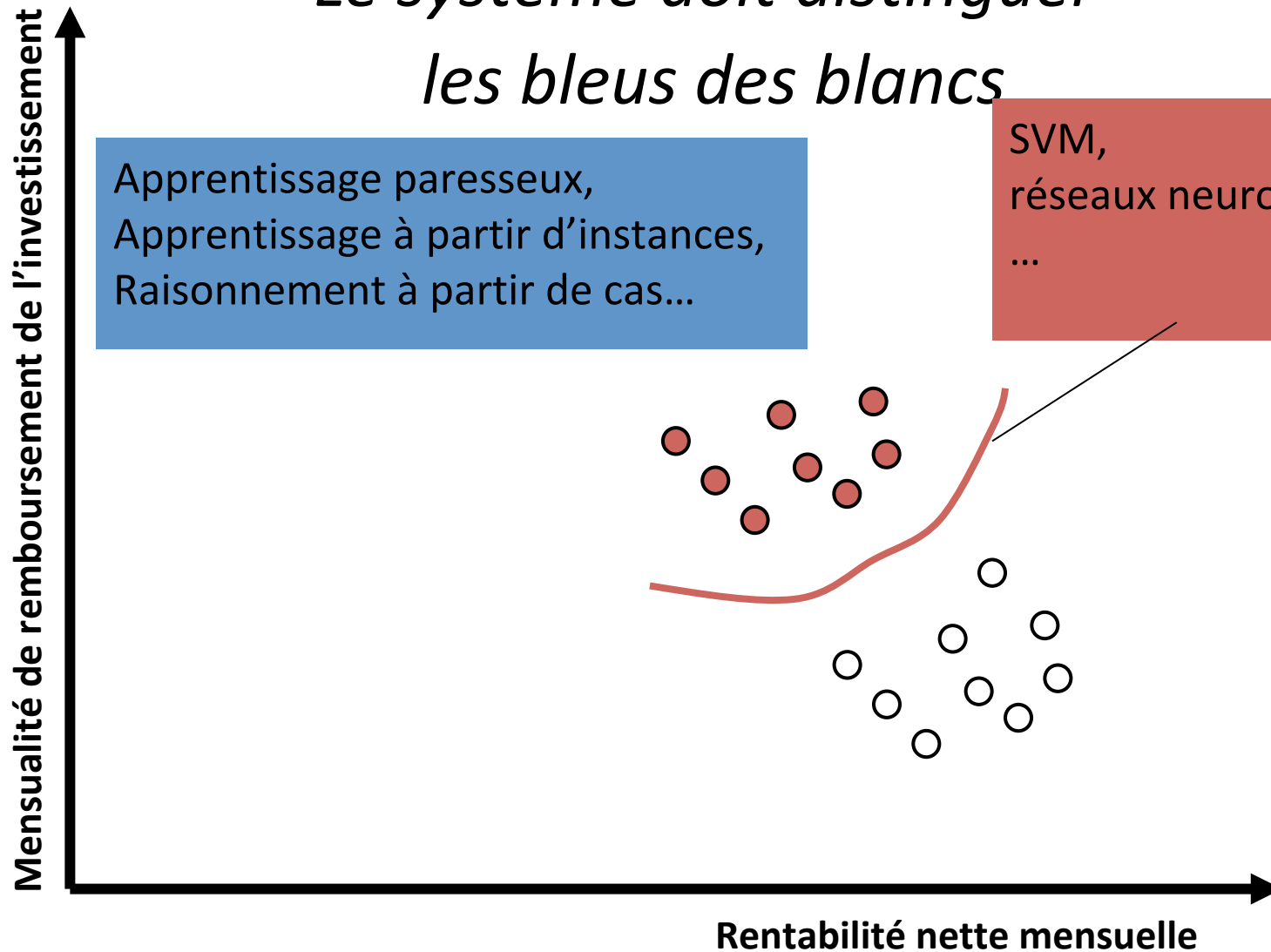
$$g(x_q) = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{i=k} w_{s(i)} \cdot \delta(v, f(x_{s(i)}))$$

- Avec: $w_i = \frac{1}{d(x_q, x_i)^2}$



Poursuite de l'apprentissage... ou non?

*Le système doit distinguer
les bleus des blancs*



Case-based Reasoning - CBR

Raisonnement à partir de cas - RàPC

- Le RàPC est fondé sur un modèle de “traitement humain de l'information” à l'intérieur d'un domaine du savoir
- Exemples de domaines: *Justice, diagnostic, médecine, planification stratégique, etc.*
- Principe: *lorsqu'ils résolvent de nouveaux problèmes les hommes font appel à leurs expériences passées*



RàPC

- Le RèPC émule le “traitement humain de l'information” fondé sur le réemploi de cas singuliers
- Deux étapes majeures
 - Retrouver, dans une base de cas, les cas pertinents pour résoudre le problème en cours d'étude
 - Concevoir la solution au problème courant en se fondant sur l'expérience passée



Analogie et RàPC

- Cas (*expérience singulière – casuistique*):
 - Raisonement à partir de cas: *raisonnement à partir d'une multitude d'expériences singulières*
 - Raisonement dans un domaine
 - Médecine, cuisine...
 - Mesure de similarité
- Analogie:
 - Raisonement du particulier au particulier
 - Passage d'un domaine à un autre
 - Syst. solaire → atome
 - Ressemblance structurelle



Méthodes à base d'instance - RàPC

À base d'instance

1. Apprentissage paresseux
 - Les instances sont conservés tels quels
2. Seules les instances proches sont utilisées
3. Instances = vecteurs de \mathbb{R}^n

À partir de cas

1. Apprentissage paresseux
 - Les cas sont conservés tels quels
2. Seuls les cas proches sont utilisés
3. **Représentation et utilisation de connaissances symboliques**

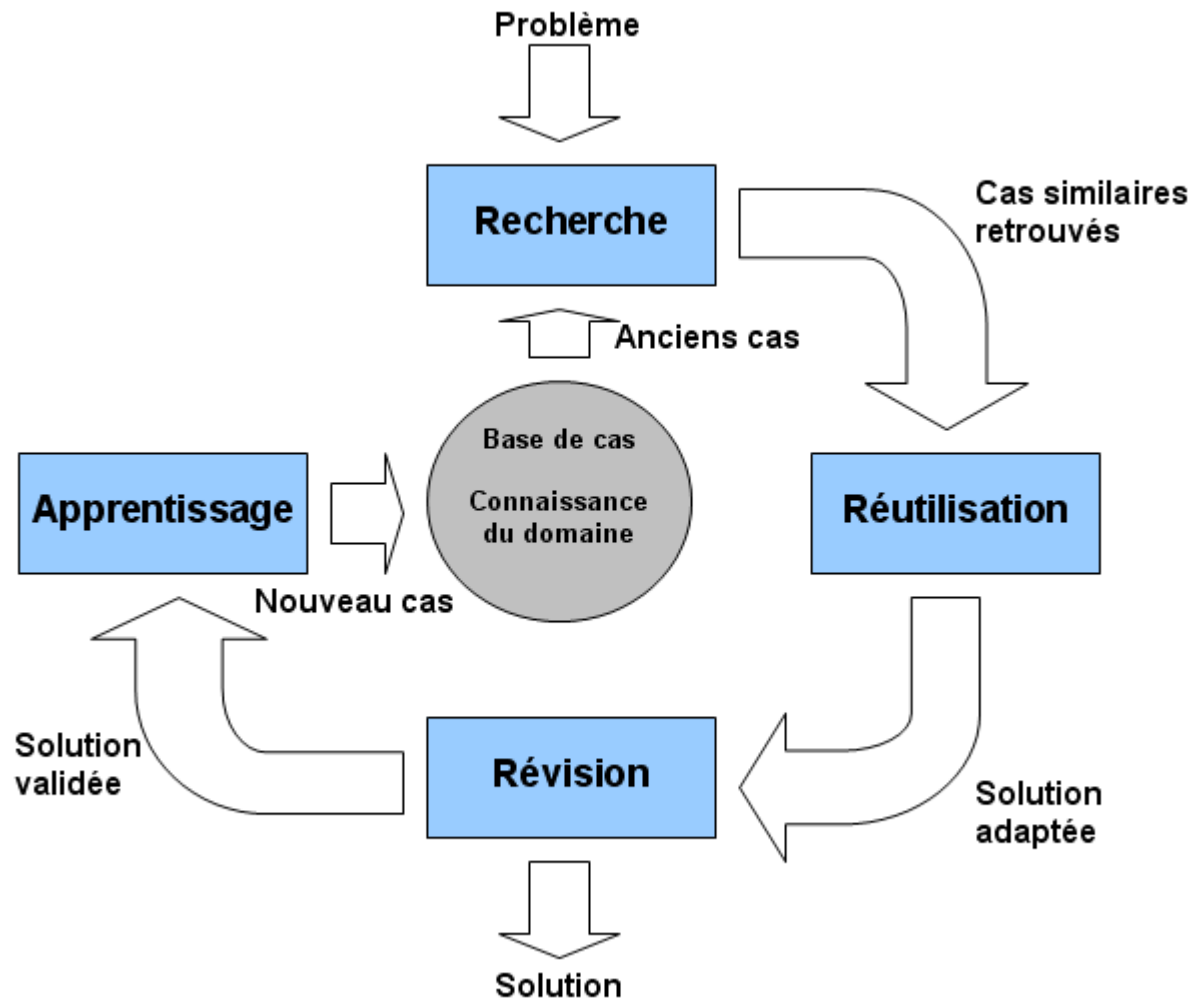


Composants du RàPC

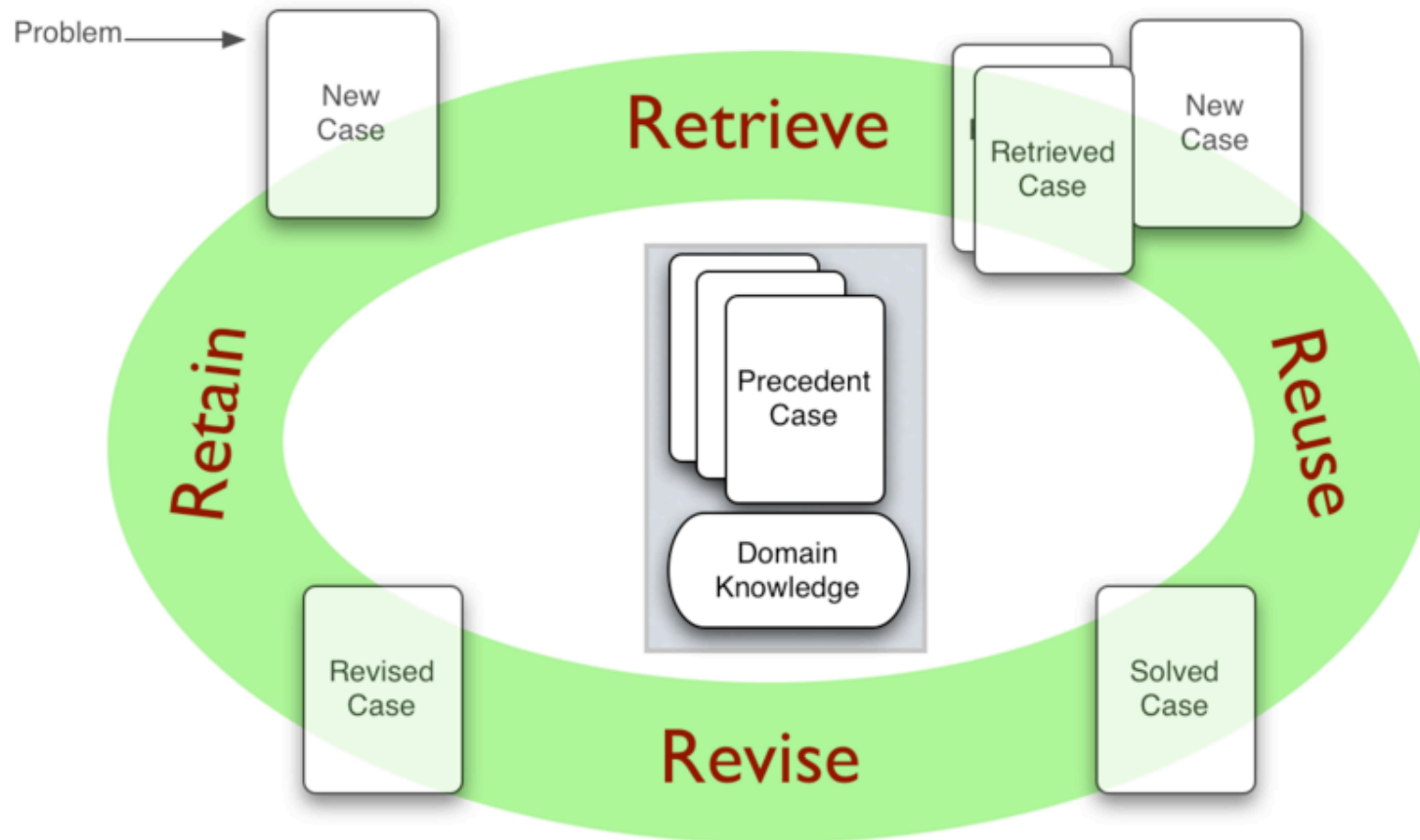
- Un système de RàPC comprend:
 - Une base de cas
 - Un trouveur
 - Un adaptateur
 - Un module de révision
 - Un module d'exécution
 - Un évaluateur



Etapes du processus



En anglais, les 4 "R"



Base de cas

- Stockage de cas antérieurs
- Les cas sont indexés de sorte qu'ils puissent être rapidement rappelés quand c'est nécessaire
- Un cas comprend une description générale de problèmes anciens



Structure d'un cas

Les cas contiennent:

- La description du problème
- La solution et les étapes qui y ont mené
- Le résultat de l'évaluation
- L'explication des échecs

Chaque cas est décrit par une liste de couples <attribut, valeur>, les attributs étant typés:

Types classiques : texte, entier, réel, booléen, date.

Type symbole : énumère une liste de symboles qui seront stockés dans un arbre.
La racine de l'arbre contiendra le symbole le plus général et les feuilles les symboles les plus spécifiques.


Type cas : référence des cas qui sont des sous parties du cas considéré.

Type formule : la valeur de cet attribut est le résultat du calcul d'une formule.

Type liste : ce type est une liste d'objets utilisant les types précédents.



Exemple de cas

Patient Ref #: 1024
Patient Name: *John Doe*
Address: *12 Elm Street*
Next of Kin: *Jane Doe*
Photo: 
Age: 53
Sex: Male
Weight: 225 lbs
Height: 5' 11"
Blood Type: A neg.
...

Traits non indexés:
Non prédictifs
Ne sont pas utilisés
pour la recherche
*Fournissent de
l'information
contextuelle aux
utilisateurs*

Traits indexés:
Prédictifs et utilisés
pour la recherche



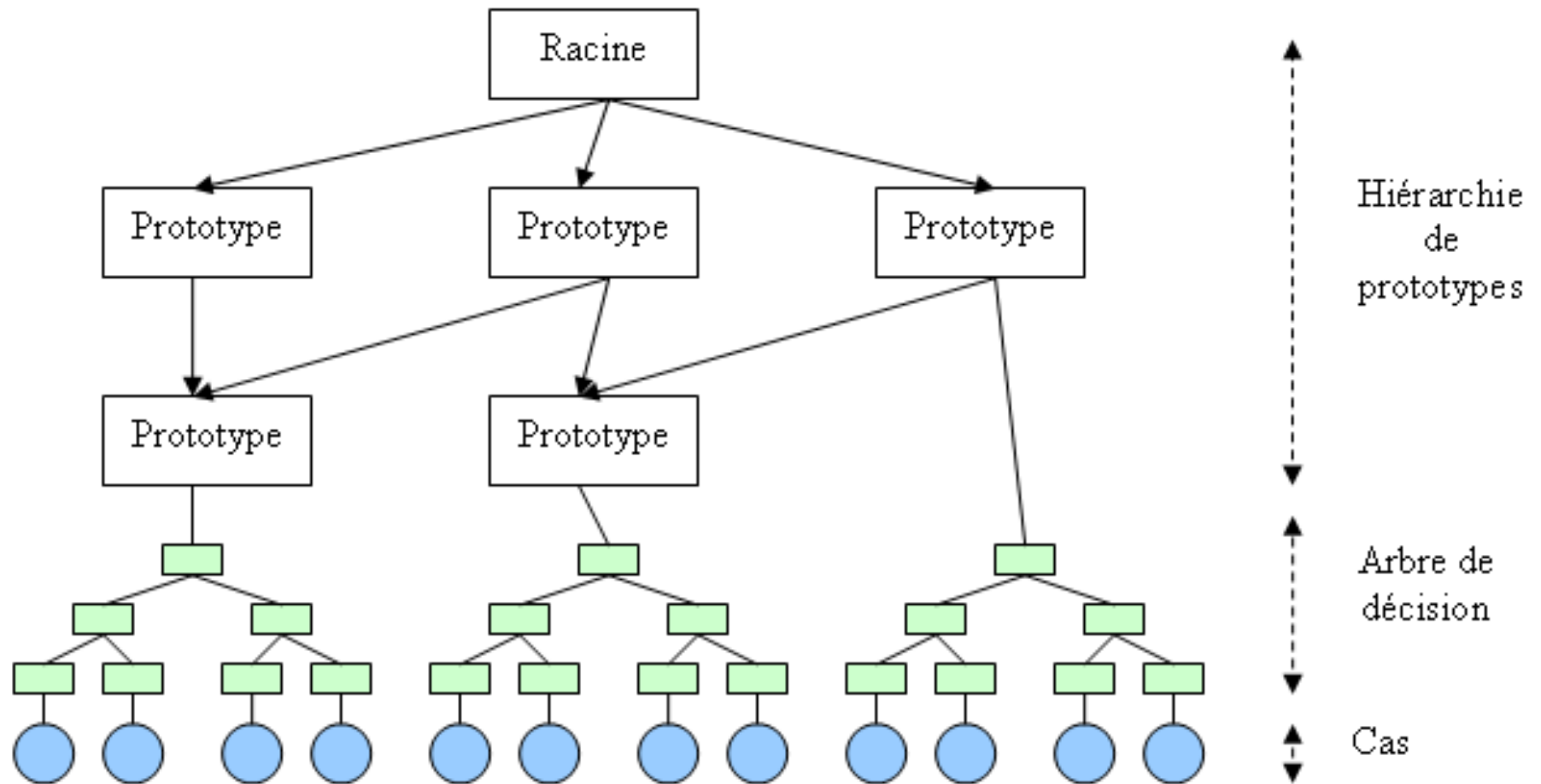
Structure de la base de cas

Organisation des cas:

- Modèle à mémoire dynamique
- Modèle à base de catégorie



Modèle simple: arbre



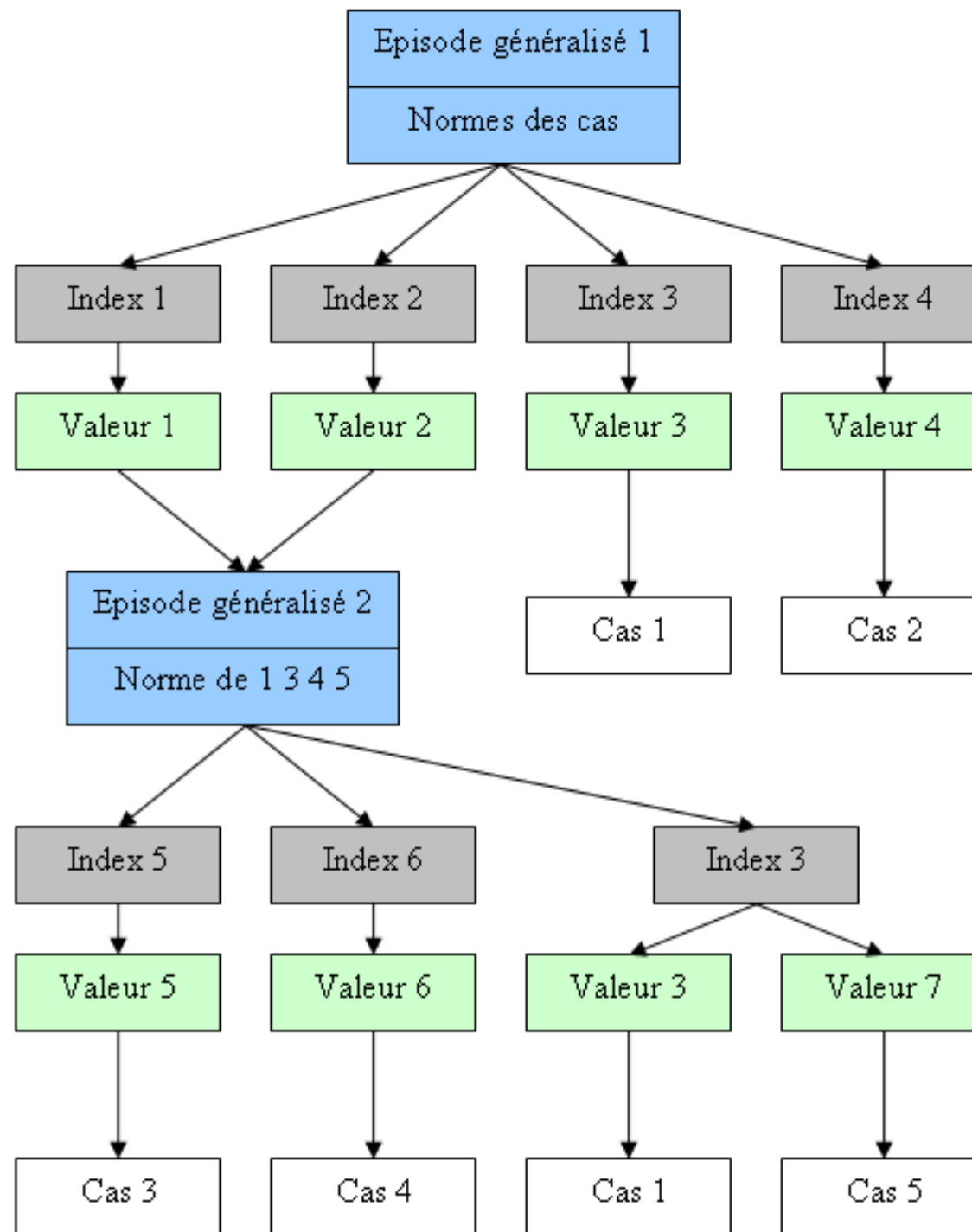
Mémoire dynamique (R. Schank)

MOP – Memory Organisation Packets

Trois types d'objets:

- **Les normes** : Les caractéristiques communes à chacun des cas indexés sous l'épisode généralisé.
- **Les index** : Les éléments discriminant les cas contenus dans l'épisode généralisé. Un index possède deux champs : son nom et sa valeur. Il peut pointer vers un autre épisode ou simplement vers un cas.
- **Les cas** : La connaissance du système. On y accède donc par l'intermédiaire d'index.





Cycle du RàPC

Les étapes du cycle du RàPC

- **Retrouver un cas dans la base de cas / Recherche**
→ Trouve les cas qui sont supposés être capables d'aider à résoudre le problème cible
- **Réutilisation**
→ Suggère une solution pour le cas cible en se fondant sur les solutions employées dans les cas trouvés; on peut éventuellement adapter les solutions des cas trouvés pour le cas cible
- **Révision**
→ Évalue les solutions choisies eu égard au succès rencontré
- **Mémorisation / Apprentissage**
→ Le résultats de la résolution de problème est intégré à la base de cas; il accroît la connaissance du système



“Trouveur”

- Quand un nouveau problème est intégré dans la base de cas, le “trouveur” décide des traits pertinents pour trouver des cas similaires
- La découverte de cas similaire recourt aux descripteurs du problème; ils agissent comme autant d'indices (et d'index) de la base de cas



Retrouver un élément dans la base de cas

Similarité

- Traits de surface: traits donnés dans la description du cas
- La découverte fondée sur la similarité fait appel aux traits de surface
- Il est inefficace de balayer toute la base de cas; c'est la raison pour laquelle une indexation pertinente est utile
 - découverte fondée sur les empreintes
 - Validation



Retrouver un cas

Alternatives à la recherche par similarités:

- Recherche guidé par l'adaptation
→ Recherche des cas qui sont le plus facile à “adapter”
- Recherche fondée sur la diversité
→ Combine une mesure de similarité avec différentes mesures pour distinguer les cas de grande similarité (exemple: médecine)
- Recherche guidée par le compromis
→ Un cas est plus acceptable qu'un autre s'il est plus proche de la question posée et s'il comporte un sous-ensemble des compromis que les autres cas comportent.



Adaptateur: réutilisation, “retailage”

- Examine les différences entre les cas retrouvés et le problème courant
- Applique des règles pour modifier les solutions des cas retrouvés afin de les adapter au problème courant



Méthodes d'adaptation

- Adaptation par substitution
 - Échange des parties de la solution retrouvée
- Adaptation par transformation
 - Change la structure de la solution retrouvée
- Adaptation générative
 - Dérive un nouvelle solution en réemployant les méthodes qui avaient été utilisées pour dériver la solution du cas retrouvé



Module de révision: attentes

- Un module de révision **critique** la solution adaptée en examinant **attentes *a priori***
- On peut procéder à une comparaison avec les attentes *a priori* des solutions similaires
- S'il y a échec (attentes répertoriées non satisfaites) pour une solution préconisée, le système décide alors que les indices sont suffisants pour suspecter l'échec de cette solution



Module d'exécution

- Une fois
 - qu'une solution a été trouvée,
 - qu'elle a été examinée et
 - qu'elle a fait l'objet d'une mise en cause (examen des attentes *a priori*)
- Un module d'exécution applique la solution révisée au problème courant



Évaluateur

- Si les résultats correspondent à ce qui est attendu, il n'y a pas d'analyse ultérieure à faire. Le cas (ou les cas) retrouvé(s) et la solution sont stockés pour être utilisés dans le futur.
- Sinon, la solution est “réparée”



Mémorisation dans le RàPC

- La forme la plus simple de mémorisation consiste à stocker le problème et sa solution comme un nouveau cas (jurisprudence)
- **Notion d'utilité:**
Lorsque la base de cas croit, tout nouveau cas n'apporte pas beaucoup d'information (recouvrement avec d'autres cas), mais augmente le temps de recherche...
- **Solution générale:**
→ Détruire les cas inutiles de la base de cas
- **Solution dans le RàPC:**
→ Utilise un modèle de compétence pour décider de ce que peut être la contribution de chaque cas à la compétence de résolution de problème du système.

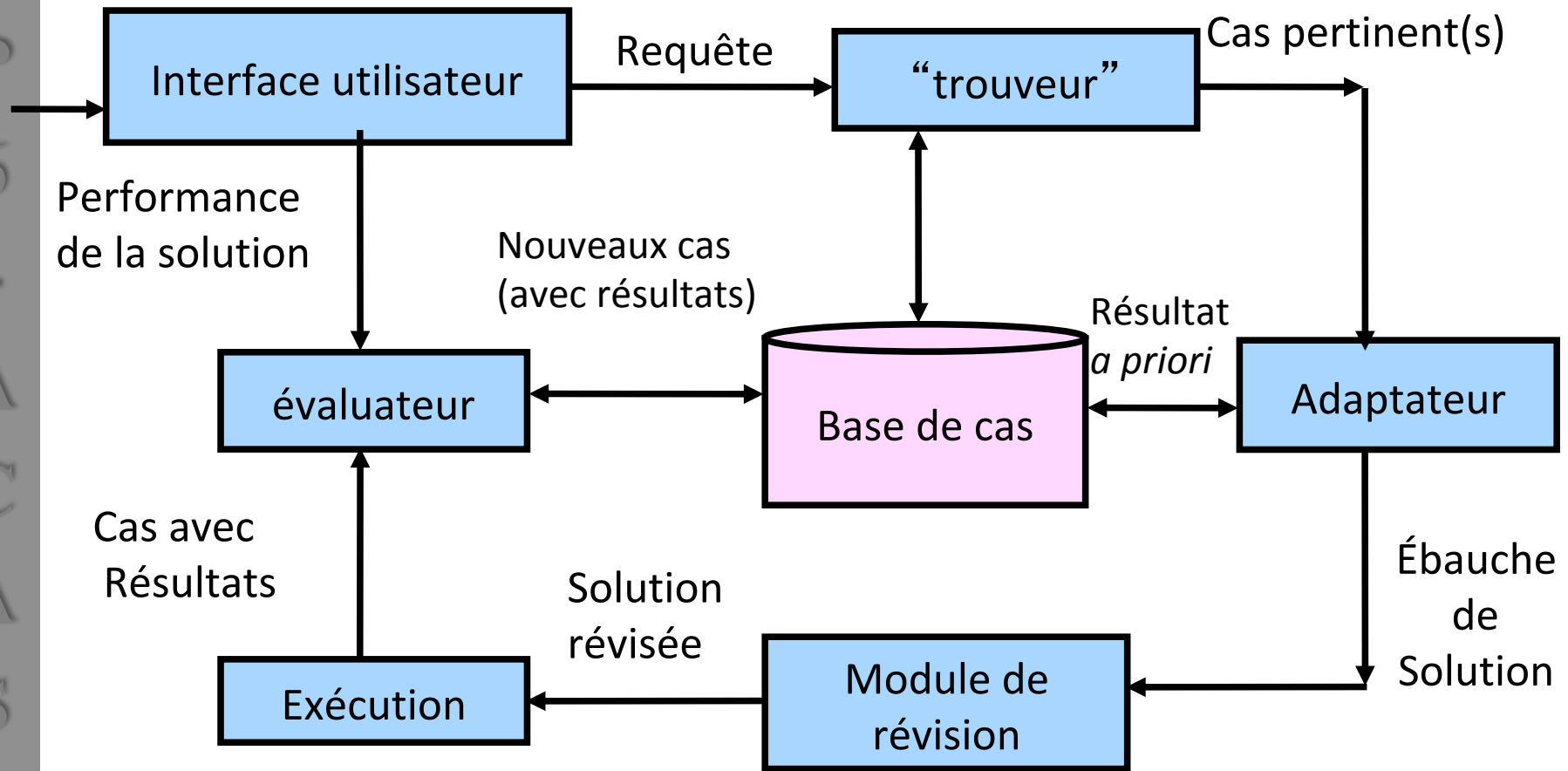


Mémorisation des cas (suite)

Maintenance de la base de cas

- Insertion de deux étapes supplémentaires dans le cycle du RàPC:
 - Revoir** – vérifie la qualité de la connaissance du système
 - Restauration** – choisit et exécute les opérations de maintenance (indexation, réorganisation, restructuration, élimination de cas inutile, etc.)
- Inventaire des questions relatives à la maintenance:
 - comment rassembler les données pour prendre les décisions de maintenance
 - comment déterminer quand déclencher des opérations de maintenance
 - type d'opérations de maintenance disponibles
 - comment exécuter les opération de maintenance





Exemple: le système CHEF *Hammond 89*

- Construction de nouvelles recettes à partir de cas représentant des recettes existantes

Une exemple de tâche à accomplir:

- Buts initiaux:
 - Faire un plat frit rapidement – c'est-à-dire un *sauté*
 - Inclure du bœuf
 - Incluse des brocolis



Retrouver des recettes semblables

- CHEF indexe la base de cas avec une hiérarchie de traits important:
 - Le type de plat est le plus important,
 - ensuite vient le type de viande,
 - ensuite les légumes, ...
- **Exemple**: si le but est de faire un **sauté de bœuf aux brocolis**, CHEF retrouve la **poêlée de bœufs aux haricots verts**.



Remarques générales sur la cuisine

On doit distinguer

- Les modes de cuisson
 - Par échanges avec ou sans coloration (*à l'étouffé en vessie, en papillote, etc.*)
 - Par saisissement avec ou sans coloration (*friture, gril, sauté, etc..*)
- Les opérations (*émincer, hacher, blanchir, déglacer, etc.*)
- Les ustensiles (*spatule, cocotte, sauteuse, etc.*)
- Les ingrédients (*viandes, poissons, épices, légumes, etc.*)



Différences culturelles

- Selon les traditions,
 - les façons de procéder,
 - les ustensiles et
 - les ingrédients diffèrent totalement



Exemple: le *wok* en cuisine chinoise



Evolutions



- Avec le temps
 - Les ingrédients évoluent (à Rome, il n’y avait ni tomate, ni polenta (car pas de maïs), ni pattes (elles viennent de Chine)...
 - Les méthodes elles mêmes changent, par exemple la nouvelle cuisine reprend les principes de la cuisine traditionnelle, mais en remplaçant la crème fraîche par du fromage blanc 0%. Une telle *substitution* est rendue possible par l’usage d’un instrument moderne, le « mixeur ».



Adaptation d'un cas

CHEF utilise deux types de connaissances d'adaptation:

- **Règles de modification** pour introduire des changements structurels (par exemple, *additions d'ingrédients* ou *réordonnancements*)
- **Etapes critiques** pour les différents ingrédients individuels



Une règle de modification

- Ajouter des fruits à un soufflé

Il faut réduire les fruits en purée et mélanger la pulpe à un produit... (crème, lait, sucre, etc.)

(add:mod

index (fruit style-souffle)

amount (cup number (1))

steps ((do (chop object ?new-item size (pulp)))

(before (pour object ?object

into (nine-inch-baking-dish))

do (mix object ?new-item with ?object))))))



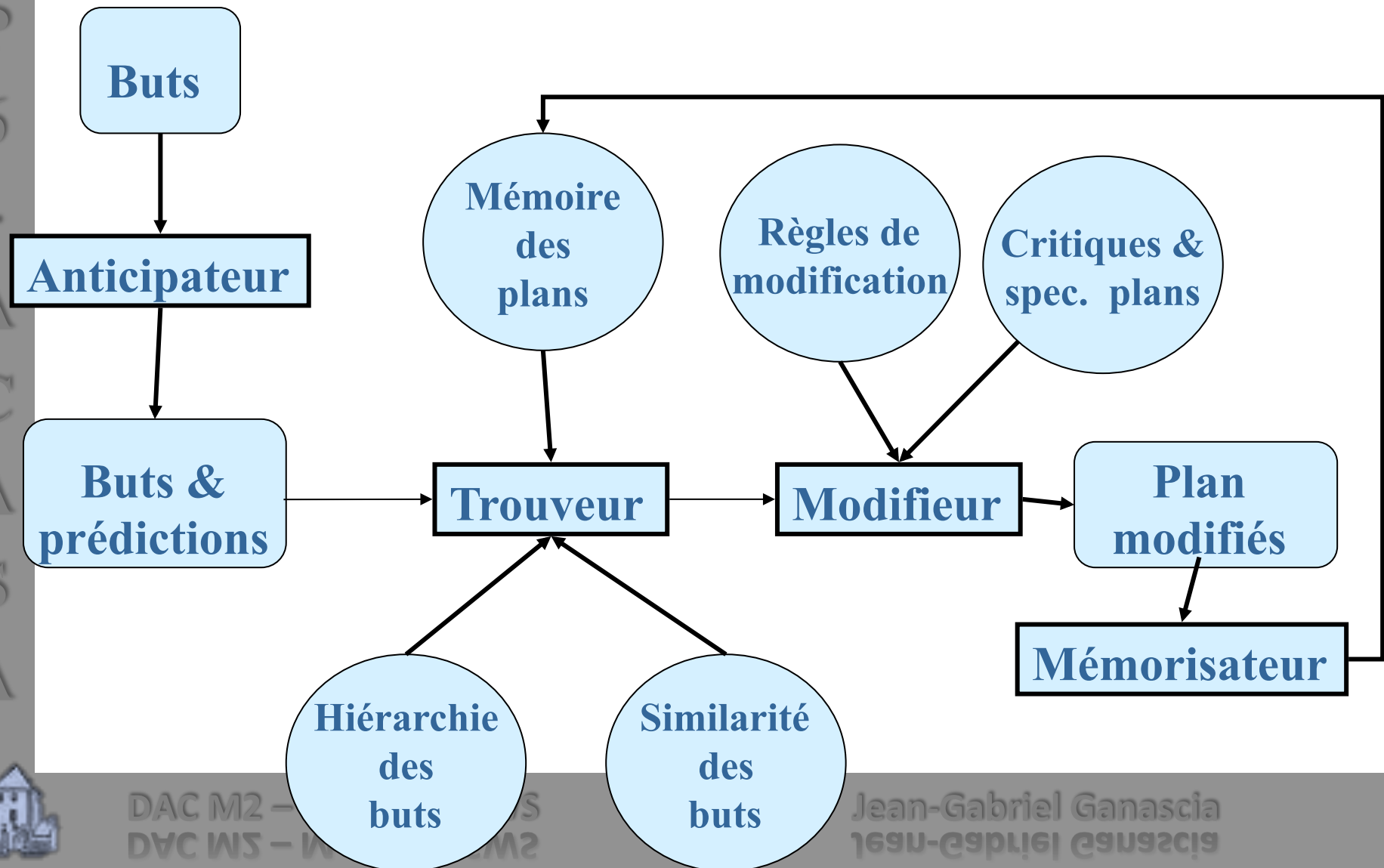
Etape critique pour les crevettes

Si on fait un plat avec des crevettes, il faut leur enlever la carapace...

```
(add:crit shrimp
  binds (shrimp *new-item*)
  steps ((before (cook-step *new-item*)
                do (shell object *new-
                    item*))))))
```



L'architecture de CHEF



Comment CHEF génère un plan pour le « sauté de bœuf aux brocolis »?

1. Trouve la recette de la poêlée de bœuf aux haricots verts (*rq. il s'agit de cuisine chinoise*)
2. Applique les règles de modification pour substituer les brocolis aux haricots verts
3. Appliquer les étapes critiques pour ajouter une étape et ajuster le temps de cuisson



Voilà, le « sauté de bœuf aux brocolis »

Ingrédients: ½ livre de bœuf, 1 cuillère de sucre, 2 cuillère de sauce de soja, ½ livre de brocolis, 1 cuillère d'alcool de riz, 1 cuillère de sel, 1 gousse d'ail

- Hacher l'ail très fin
- Émincer le bœuf
- Faire mariner le bœuf dans l'ail, le sucre, l'alcool de riz et la sauce de soja
- Couper les brocolis en morceaux
- Faire revenir le bœuf avec les épices et l'alcool de riz pendant une minute
- Ajouter les brocolis au bœuf
- Faire sauter le tout pendant trois minutes
- Ajouter le sel



Evaluation du résultat

- Confronter le résultat avec les objectifs de la recette:
 - Le bœuf doit être tendre
 - Le plat doit avoir une saveur satisfaisante
 - Les brocolis doivent être fermes
 - Le plat est salé
 - Le plat est un peu sucré
 - Le goût du plat doit laisser ressortir l'ail
- Le plan est exécuté dans un simulateur et on vérifie qu'il satisfait les attentes

Mais le brocoli est trempé!

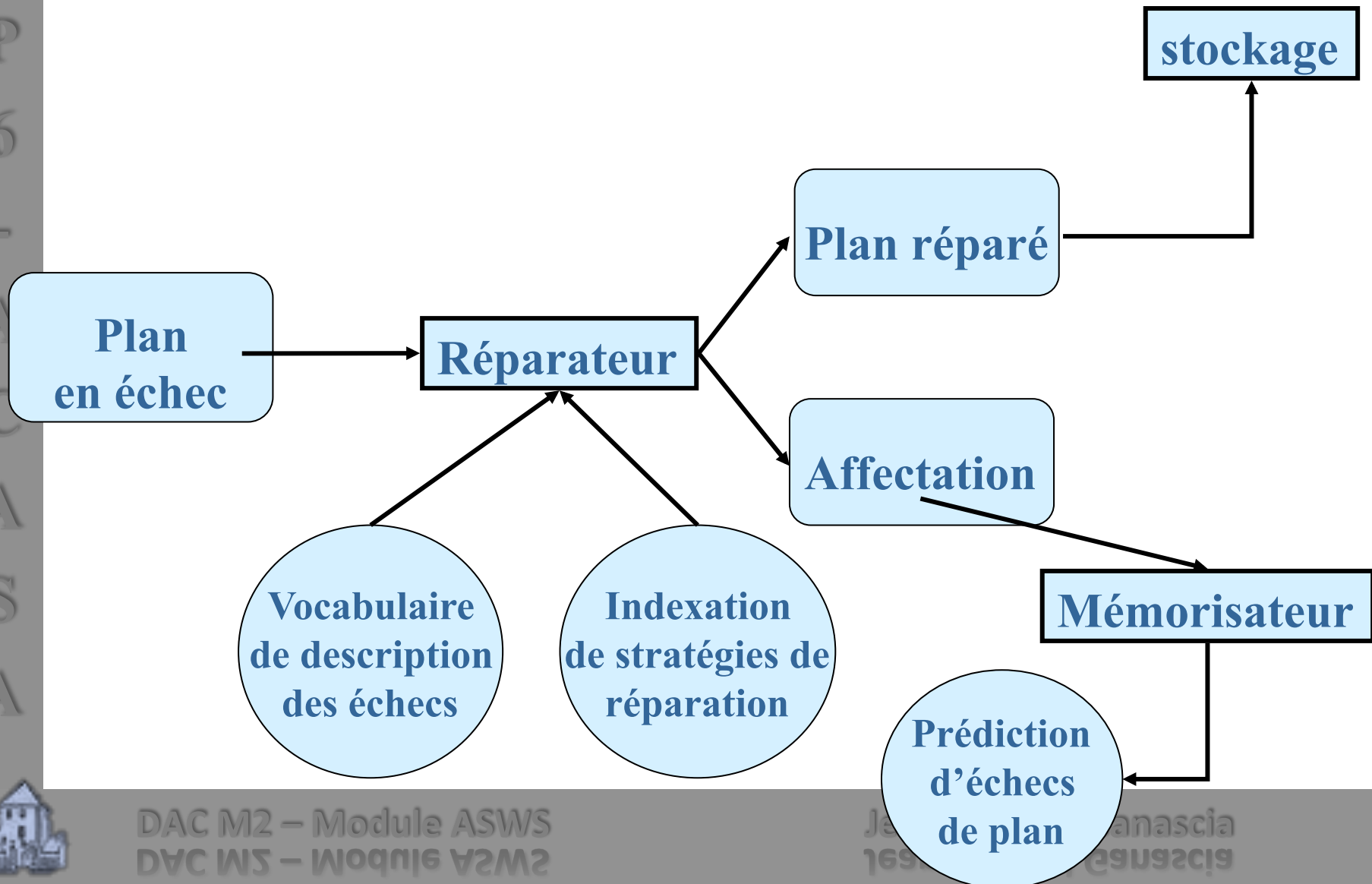


Récupération des échecs par CHEF

- Explication: le bœuf donne de l'eau → les brocolis cuisent dans la vapeur
- Identifier le motif du problème:
 - L'eau relâchée lors de la cuisson du bœuf ne permet pas un « wok sec »
 - Effet de bord: ne permet pas la condition courante
- Retrouver les stratégies de réparation à partir de la forme du problème
 - Diviser et modifier
 - Altérer le plan en fonction des effets de bord
- Appliquer les stratégies « diviser et modifier »
- Le nouveau plan marche!



Récupération des échecs



Apprentissage par expérience

- CHEF stocke les plans sous les combinaisons de buts qu'ils satisfont, incluant les problèmes qu'ils évitent
- Il apprend:
 - Un nouveau cas: la recette du bœuf aux brocolis indexée sous:
 - Buts (sauté, bœuf, brocolis)
 - Les problèmes évités (éviter les mauvaises interaction entre viande et légumes)
 - Une règle pour anticiper les problèmes d'interaction entre les viandes et les légumes sautés.





9th European
Conference on
Case-Based Reasoning
September 1st - 4th, 2008, Trier, Germany



[Welcome](#) [Call for Papers](#) [Submission](#) [CCC](#) [Programme](#) [People](#) [Industry Day](#) [Workshops](#)

- » CCC Call
- » Exercise Material
- » Developers Wiki
- » Public CBR Tools
- » Competition Rules
- » Timeline
- » Grants
- » Newsletter CCC
- » Contact CCC



[Call in pdf](#)

CALL FOR PARTICIPATION

1st Computer Cooking Contest

CCC 2008 @ ECCBR 2008

September 1, 2008, Trier (Germany)

www.computercookingcontest.net

Who says that only human beings are able to cook delicious meals? We aim to teach our computers the haute cuisine and therefore we need your creativity and ideas! Come to the ECCBR'08 in Trier and participate in the Computer Cooking Contest (CCC)! Write your own software application for the live competition that will take place at September 1, 2008. Show that your program is more creative than the average kitchen user. Let your computer's recipe creations be evaluated by a professional cook and an international jury of scientists.



Universität Trier





9th European Conference on Case-Based Reasoning

September 1st - 4th, 2008, Trier, Germany



[Welcome](#) [Call for Papers](#) [Submission](#) [CCC](#) [Programme](#) [People](#) [Industry Day](#) [Workshops](#)

- » [CCC Champions](#)
- » [CCC Finale](#)
- » [CCC Call](#)
- » [CCC Submission](#)
- » [Developers Zone](#)
- » [Competition Rules](#)
- » [Timeline](#)
- » [Press](#)
- » [Jury Members](#)



Champions of the 1st Computer Cooking Contest

- **European Champion on Computer Cooking: Team *What's in the fridge?*** from Dublin Institute of Technology, Ireland
Qian Zhang, Rong Hu, Brian Mac Namee, and Sarah Jane Delany
- **European Vice Champion on Computer Cooking: Team *TAAABLE*** from three Universities in France
Fadi Badra, Rokia Bendaoud, Rim Bentebitel, Pierre-Antoine Champin, Julien Cojan, Amélie Cordier, Sylvie Després, Stéphanie Jean-Daubias, Jean Lieber, Thomas Meilender, Alain Mille, Emmanuel Nauer, Amedeo Napoli, Yannick Toussaint
- **Champion of the Negation Challenge on Computer Cooking: Team *ColibriCook*** from Complutense University Madrid, Spain
Juan DeMiguel, Laura Plaza, and Belén Diaz-Agudo
- **Champion of the Menue Challenge on Computer Cooking: Team *CCCIIS*** from



Universität Trier



DAC M2 – Module ASWS
DVC WS – W0916 V2M2

Jean-Gabriel Ganascia
169U-@9PLIG @9U92CIS

Taaable

Ingredients

I want:

 ?

I don't want:

 ?

Type of dish

I want:

 ?

I don't want:

 ?

More options

Vegetarian Nut-free No alcohol

[Advanced Configuration ?](#)

Find recipes!

Get 5!

Reset query



More options

Vegetarian Nut-free No alcohol

[Advanced Configuration ?](#)

Find recipes!

Get 5!

Reset query

Your request: **veal mushroom D:french**

Common path: 1 <meat --> veal >2 <food --> mushroom >

Common cost: 11.878344607290492

#	Original recipe name	Adaptation overview
1	Beef Stew	Replace: D:meat by meat, beef chuck by meat, carrot by vegetable, stalk_celery by vegetable, potato by vegetable onion by vegetable, garlic by vegetable

Results 1 - 1 on 1 | Processing time: 1.9684 secondes

Taaable

[Help me!](#) | [More about Taaable](#) | [The team](#) | [Administration](#)

Taaable 1.2 - Copyleft Taaable Team 2008

Terminé

Maintenant : Nuageux, 8°C

Lun : 8°C

Mar : 8°C

Mer : 9°C

Jeu : 10°C



DAC M2 – Module ASWS
DVC WS – Wobnig A2M2

Jean-Gabriel Ganascia
169U-@9P1IGI @9U92CIS

Beef Stew

Type(s): D:europa,D:french,D:main dish,D:stew,D:meat

Substitution List

- ◆ Replace **beef_chuck, D:meat**, by **veal**
- ◆ Replace by
- ◆ Replace by **mushroom**

Original Recipe

Ingredients :

Original ingredient	Prefered term
3 carrots, sliced	carrot
3 potatoes, diced	potato
2 lb beef chuck or stew meat, cut in cubes	beef chuck



Taaable

Ingredients

I want:

fish_fillet carrots bean ?

I don't want:

?

Type of dish

I want:

swedish ?

I don't want:

?

More options

Vegetarian

Nut-free

No alcohol

[Advanced Configuration ?](#)

Find recipes!

Get 5!

Reset query



L

Your request: **fish_fillet bean D:swedish**

Warning, the following terms were ignored: **carrots**

Common path: 1 <food --> not_vegetarian>2<legume --> bean>3<food --> fish_fillet>

Common cost: 19.5152292371289

#	Original recipe name	Adaptation overview
1	Swedish Cheese Pie	Replace: almond by legume

Results 1 - 1 on 1 | Processing time: 2.2304 secondes

Taaable

A

S

A





Swedish Cheese Pie

Type(s): D:baked good,D:dessert,D:pie,D:swedish,D:sweet

Original Recipe

Ingredients :

- 1 x basic pastry pie crust; 9"
- 2 c cottage cheese
- 3 ea eggs; large
- 1/4 c unbleached flour; sifted
- 1/4 c sugar; granulated
- 1 c light cream
- 1/2 c almonds; toasted, fine chop

Recipe :

preheat the oven to 350 degrees f press the cottage cheese through a sieve place in a large mixing bowl and beat until smooth add the egg flour sugar cream and finely chopped almond and blend well pour the mixture into the prepared crust and bake for about 45 minutes or until a knife comes out clean remove the pie from the oven and chill before serving.



Computer Cooking Contest 2017

An event of the International Conference on Case-Based Reasoning (ICCBR 2017)

Home

An event of the International Conference on Case-Based Reasoning (ICCBR 2017)

Goals of the Computer Cooking Contest

Once upon a time, we wondered whether some software system could help us to make a yummy meal from the contents of our fridge. Given a restricted set of ingredients, the task is to cook something that tastes good. More recently, we wondered whether a system could help us to explain our research interests to a broader audience. Given the technological state-of-the-art, the task is to create a problem-solving system. Glue the two together and you get: The Computer Cooking Contest!





Logiciel CBR



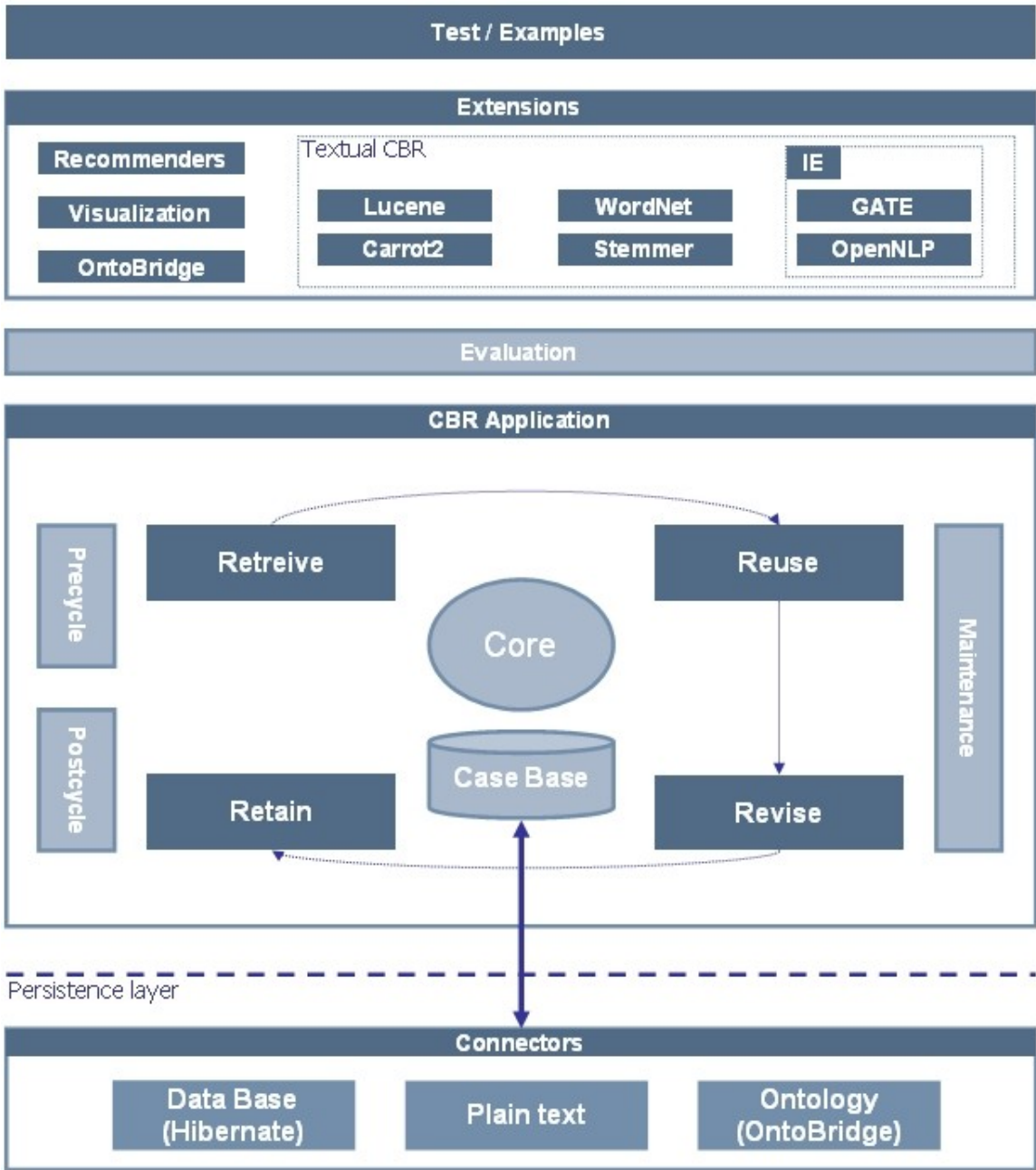
- <http://gaia.fdi.ucm.es/research/colibri/jcolibri>
- Beaucoup d'applications...
- « Poetry Generation in COLIBRI »



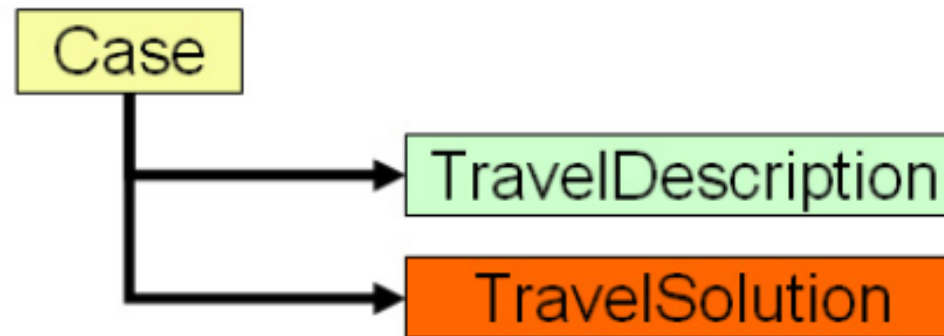
Etapes - jColibri

Les quatre R:

- Recherche
- Réutilisation
- Révision
- Retenir



Un exemple: conseiller touristique



caseld	Holiday Type	Price	NumberOf Persons	Region	Transportation	Duration	Season	Accommodation	Hotel
Journey1	Bathing	2498	2	Egypt	Plane	14	April	TwoStars	White House
Journey2	Bathing	3066	3	Paris	Plane	7	July	FourStars	Ritch

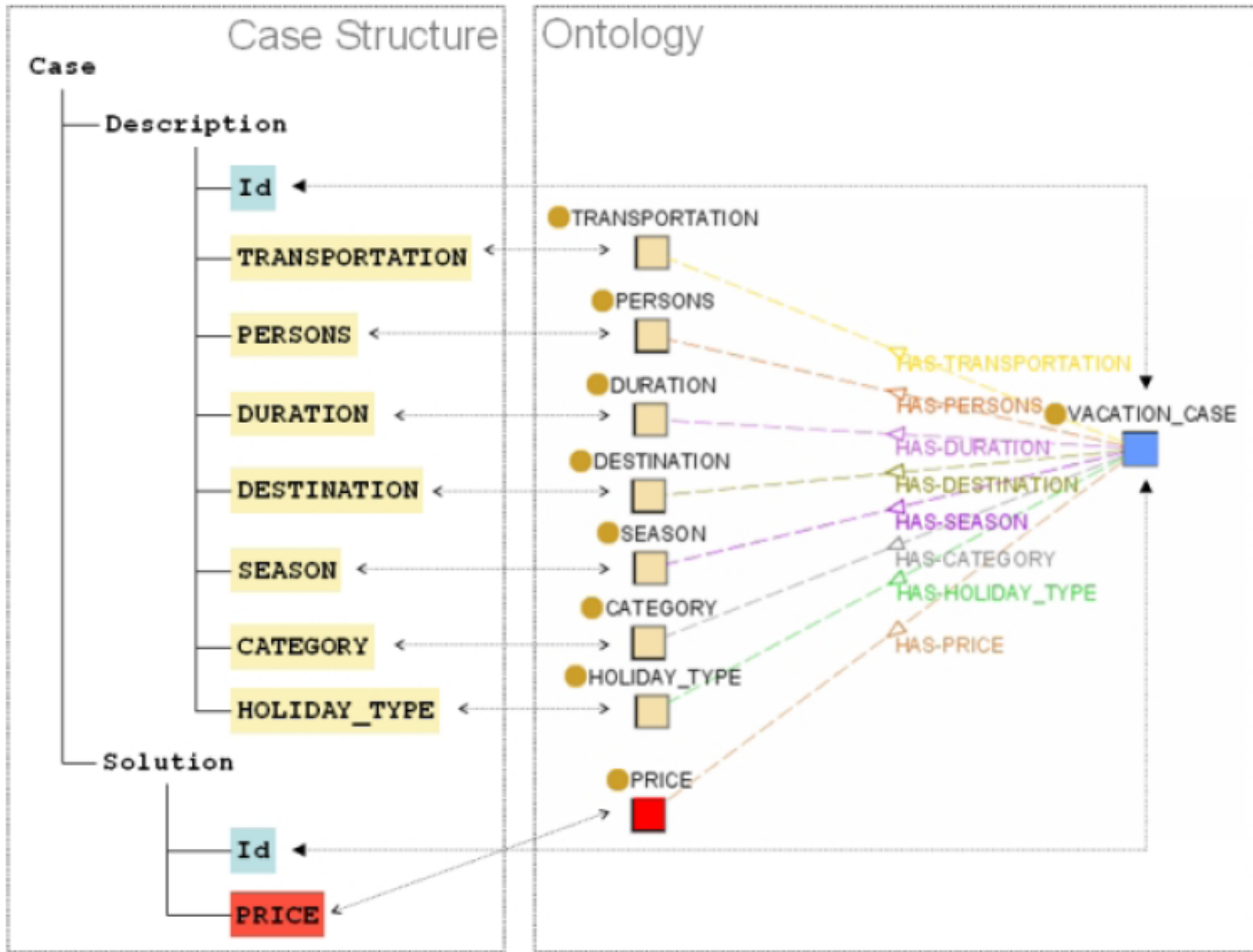


Description des cas

```
create database travel;
use travel;
drop table travel;
create table travel(caseId VARCHAR(15), HolidayType VARCHAR(20),
Price INTEGER, NumberOfPersons INTEGER, Region
VARCHAR(30), Transportation VARCHAR(30), Duration INTEGER,
Season VARCHAR(30), Accommodation VARCHAR(30), Hotel
VARCHAR(50));
insert into travel values("Journey1", "Bathing",
2498,2, "Egypt", "Plane",14,
"April", "TwoStars", "Hotel White House, Egypt");
insert into travel values("Journey2", "Bathing",
3066,3, "Egypt", "Plane",21,
"May", "TwoStars", "Hotel White House, Egypt");
```



Connaissance: ontologie



L
I
P
6
-
A
C
A
S
A



Utilisation d'une ontologie en OWL caractérisation des destinations

```
<owl:Class rdf:ID="Balearic_Islands">  
  <rdfs:subClassOf>  
    <owl:Class rdf:ID="Mediterranean"/>  
  </rdfs:subClassOf>  
  <rdfs:subClassOf>  
    <owl:Class rdf:ID="Island"/>  
  </rdfs:subClassOf>  
  <rdfs:subClassOf>  
    <owl:Class rdf:ID="Spain_country"/>  
  </rdfs:subClassOf>  
</owl:Class>
```



