

From Case-Based Reasoning to Traced Experience Based Reasoning

An introduction

Alain Mille
SILEX (Supporting Interactions & Learning by Experiences)
LIRIS CNRS 5205
University Lyon1



Summary

- Case-Based Reasoning ?
 - ◆ Genesis
 - ◆ CBR Reasoning Cycle: knowledge and reasoning processes.
- A step further : TEBR
 - ◆ TEBR cycle
- Examples
- Discussion

Alain Mille 2



Genesis...

- Marvin Minsky [1975] a model of memory -> frames
- Robert Schank [1982], a dynamic memory -> scripts
- CBR cycle [1994], CBR step by step.

Alain Mille 3

Minsky theory

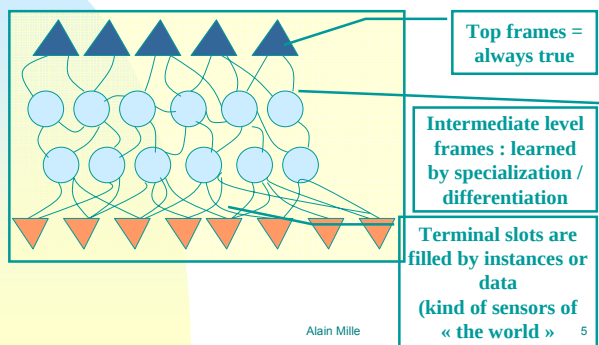
*“Here is the essence of the theory: When one encounters a **new situation** (or makes a substantial change in one's view of the present problem) one selects from memory a structure called a **Frame**. This is a **remembered framework to be adapted to fit reality by changing details as necessary.**”*

⇔ Frames are idealistic stereotypes of encountered situations....

Alain Mille

4

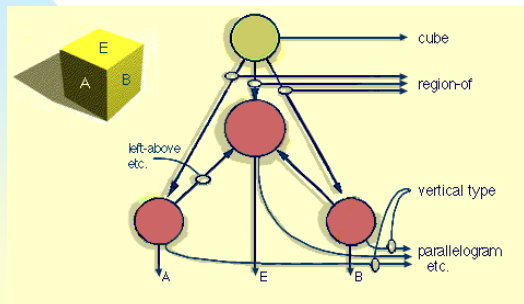
Minsky : the memory



Alain Mille

5

A specific frame



Alain Mille

6

Minsky, the process

- Frames are organized according to a « specialization » hierarchy and are connected by « proximities » links
- Process :
 - ◆ Select a frame
 - ◆ Apply the frame in the context of the current situation, balancing failed and reached goals.
 - ◆ Adapt/correct frame application in order to fit the current situation
 - ◆ Synthesize this experience in the memory framework.

Alain Mille

7

Schank : dynamic memory

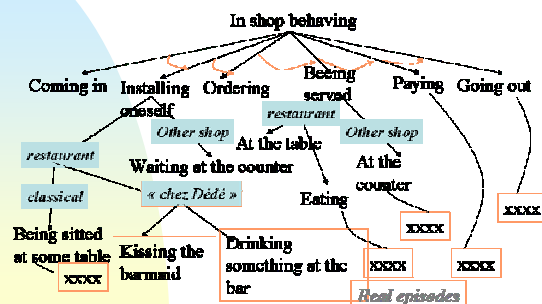
- Understanding ⇔ Explaining
 - ◆ Natural language understanding.
 - ◆ Using scripts to describe propositions
 - ◆ Using concrete experience to build incrementally the scripts.

“John went to the restaurant. He got some ham. It was good.”.

Alain Mille

8

Scripts memory



Alain Mille

9

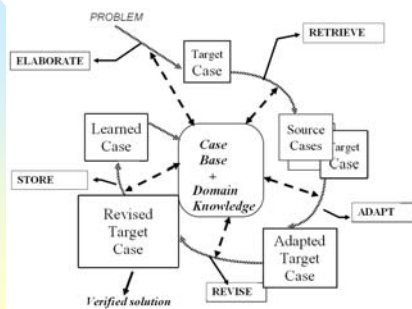
Schank: reasoning process

- In a memory of concrete experiences (dynamically organized in a generalization hierarchy),
- One retrieves the « closest » experience to the current one,
- The corresponding script has to be generalized as less as possible to be re-specialized in the current context,
- Memory is re-organized if necessary according to what happens really.

Alain Mille

10

Case-Based Reasoning Cycle



Alain Mille

11

What is a case?

- A case is the description of a problem solving episode. Its structure fits the situation of the task: diagnostic, planning, decision helping, design, etc.

Alain Mille

12

Case descriptors

- A case is composed of a problem description and its corresponding solution description: $cas=(pb,Sol(pb))$.
- A **source** case is a case whose solution will be adapted to find a solution for a new case, we call **target case**. A source case is formulated as:
 - ♦ $Source_case=(source,Sol(source))$
- and a target case:
 - ♦ $Target_case=(target,Sol(target))$.
- A case, its problem and solution are described by a set of descriptors. A descriptor d is defined by a pair $d=(a,v)$ where 'a' is an attribute name and 'v' its value in a case.

Alain Mille

13

Descriptors

- $source=\{d_1^s..d_n^s\}$ where d_i^s is a source problem descriptor.
- $Sol(source)=\{D_1^s..D_m^s\}$ where D_i^s is a source solution descriptor
- $target=\{d_1^t..d_n^t\}$ where d_i^t is a target problem descriptor.
- $Sol(target)=\{D_1^t..D_n^t\}$ where D_i^t is a target solution descriptor.

Alain Mille

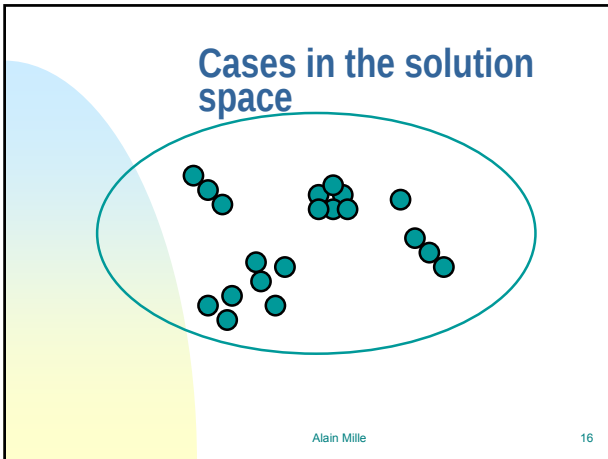
14

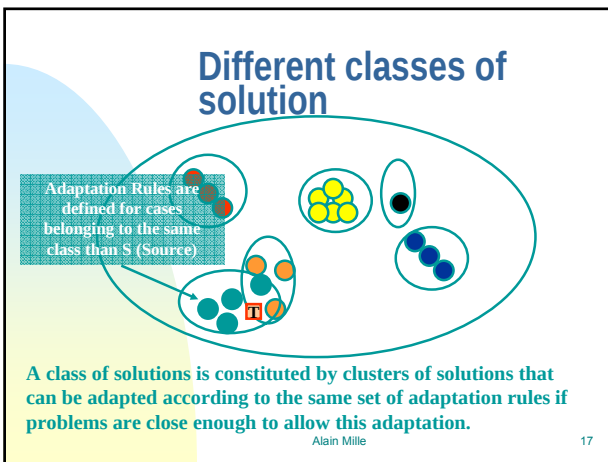
Case_base (example)

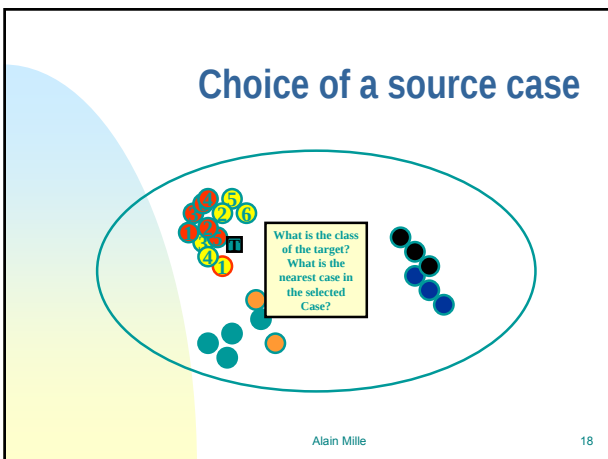
Attribute label	Case 1	Case 2	Case 3	Attribute type
Pb_Surface	55	35	55	Real
Pb_District_Location	Rhône district	Rhône district	Ain district	Symbol
Sol_Sale_Price	20000	45000	15000	Real
Pb_Flat_Type	F2	F2	F2	Symbol
Pb_Town_Location	Lyon	Lyon	Bourg en Bresse	Symbol

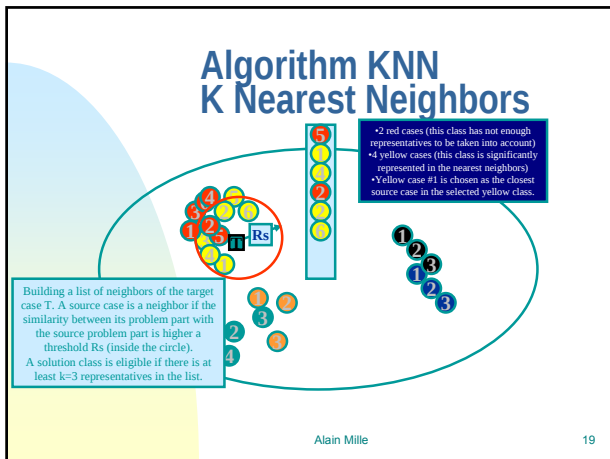
Alain Mille

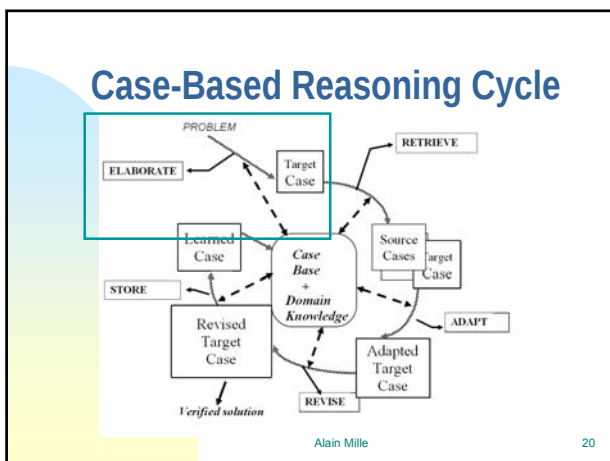
15











Elaborate

- Warning : we search for similar **solutions(!)** by using similarities on **problem parts...**
 - ◆ **Supplementing/filtering** problem description on the basis of available knowledge about **adaptability**.
 - ◆ Trying to solve the problem... -> generation of a set of problem descriptors maximizing chances to find useful source cases.

⇒ **Guiding the retrieval** of an **adaptable** solution

Alain Mille 21

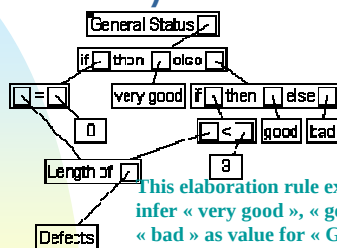
Elaborate: illustration

- Cases are car « selling episodes »
- Problem is constituted of descriptors of the car
- Solution has a unique descriptor: the price of the car

Alain Mille

22

Domain knowledge (part of...)



This elaboration rule expresses how to infer « very good », « good » or « bad » as value for « General Status » descriptor by simply counting the number of listed defects in the Defects descriptor.

Alain Mille

23

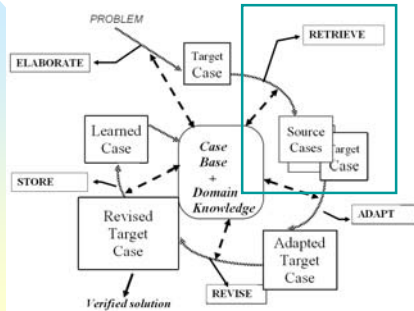
Car descriptors (for sale!)

Att label	Att type	Att-value	Elaborated value
General status	Symbol (inferred)	??	Good
Nb kms	Real	198000	198000
Nb of years of the car	Real	10	10
Car Manufacturer	Symbol (inferred)	??	Pengoot
Car model	Symbol	206	206
Car type	Symbol	Break	Break
Defects	List of symbols	(superficial problems)	(superficial problems)
Sale Price (solution)	Real	???	???

Alain Mille

24

Case-Based Reasoning Cycle



Alain Mille

25

Retrieve

- Similarity = matching degree between two problems (problem parts of cases)
 - ◆ Eliciting descriptors to match
 - ◆ Computing local similarity on matched descriptors
 - ◆ Computing global similarity by some « weighted » function.
 - ◆ (Building a similarity path in order to guide a corresponding adaptation path)

Alain Mille

26

Example of a simple similarity measure

$$Similarity_{case}(Target, Source) = \frac{(\sum w_i * Similarity_{descriptor}(d_i^t, d_i^s))}{\sum w_i}$$

w_i : weight of the descriptor d_i

w_i the « weight » of the case descriptor #i expresses the « difficulty to adapt » the solution of the case regarding the difference between the corresponding source and target descriptors #i.

Alain Mille

27

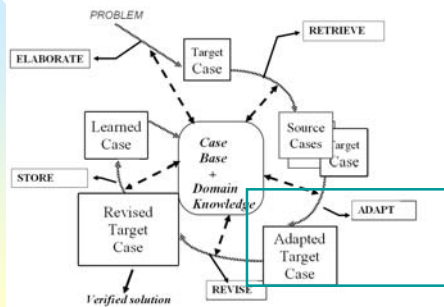
An example of weights (importance) of attributes / price of a car

Attribute label	Attribute type	Influence <i>weight</i> of the attribute on the solution
General Status	Symbol (inferred)	20%
Nb of kms	Real	35%
Nb of years of the car	Real	25%
Manufacturer	Symbol (inferred)	5%
Car Model	Symbol	5%
Car type	Symbol	10%
Observed defects list	List of symbols	No importance
Sale Price (solution)	Real	???

Alain Mille

28

Case-Based Reasoning Cycle



Alain Mille

29

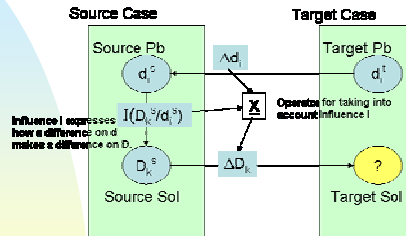
Adapt : the central step in CBR

- The question is to reuse the solution of a close source case,
- Because it is possible to adapt this source solution for the close target problem,
- ... and that it is easier to adapt a close source solution than to solve the target problem from scratch!

Alain Mille

30

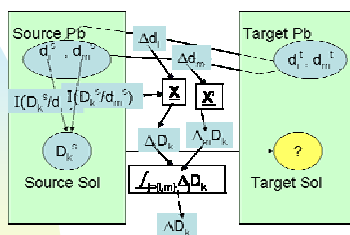
Adapt : formalization



Alain Mille

31

Adap : formalization



Alain Mille

32

Adaptation : formalization

- Δd_i = discrepancy between source and target problem descriptors values according to a specific matching function.
- $I(D_k^s/d_i^s)$ = influence of a discrepancy of d_i^s on the value of D_k^s
- $\otimes$ = operator to compute Influence according to the observed problem descriptors discrepancies.
- $\int_{i=1, m} (I(D_k^s/d_i^s) \times \Delta d_i)$ sums individual influence effects of problem descriptors discrepancies for an "individual" source solution descriptor (there is a simple generalizing equation for several source solution descriptors).
- $\oplus$ = operator of "addition" of the integrated computed influence to a source solution descriptor to propose a value for the corresponding target solution descriptor

Alain Mille

33

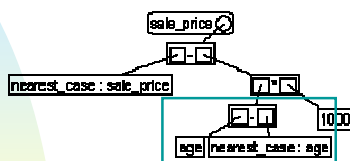
Adaptation Example : Similarity Knowledge

MODE:		IMPORTANCE LEVEL:	
☐ Inductive		☐ Exact	☐ Ignore
☒ Nearest Neighb		☒ 1	☐ 2 ☐ 4 ☐ 8 ☐ 16
☐ Adaptation		Global Weight Vector : Default W	
Field Name	Field Value	Field Type	
age	16 (28%)	Integer	
color	1 (2%)	Symbol	
Defects		List of Symbol	
gas	8 (14%)	Symbol	
General Status		Symbol (Formula)	
km	8 (14%)	Integer	
model	8 (14%)	Symbol	
nearest_case		Case	
sale_price		Real	
Trade name	16 (28%)	Symbol	

Alain Mille

34

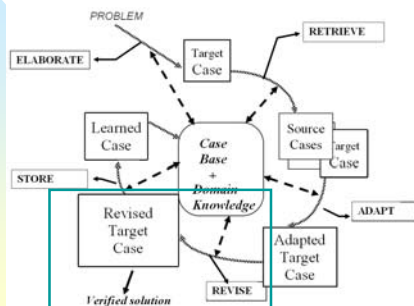
Adaptation example: adaptation rule (a very simple one!)



Alain Mille

35

Case-Based Reasoning Cycle



Alain Mille

36

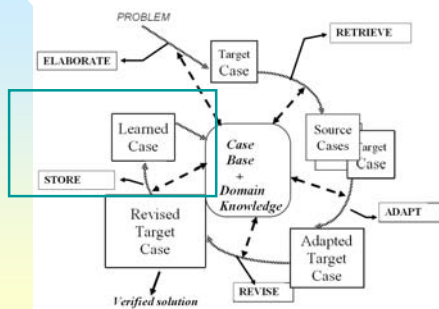
Revise

- In order to revise, it is possible to:
 - ◆ Check the adapted solution « against » the real world...
 - ◆ Introspect the Case Base with all descriptors of the adapted case -> is there similar past cases which were not working this way?
 - ◆ Use another problem solving process (expert system, simulator, ...)
- Revising allows to learn from errors !

Alain Mille

37

Case-Based Reasoning Cycle



Alain Mille

38

Memorize

- Adding a new real solved case to the Case base is the basic "learning" mechanism of CBR.
- Other things can be learned:
 - ◆ By interacting at "revise" step in order to manage retrieval and adaptation knowledge:
 - Similarity measure
 - Influence knowledge
 - New dependencies, etc
 - ◆ By keeping the "trace" of the "reasoning process" as it was done for the current new case. For example, if we keep trace of the adaptation process, we can consider these adaptation traces as "adaptation cases" usable in a CBR cycle for improving adaptation knowledge.

Alain Mille

39

Managing Case Bases

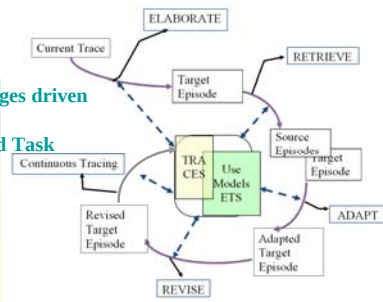
- Indexing cases
- Maintaining
 - ◆ Deleting « useless » cases
 - ◆ Managing similarity measures and adaptation rules
- Integrating Case Bases in Information Systems

Alain Mille

40

A step further... Traced Experience Based Reasoning (TEBR)

Use model: usages driven
« ontology »
ETS: Explained Task
Signature



Alain Mille

41

Our Success Stories

- Design helping of rubber production processes (Interp company -> 24 months to 8 months)
- Supervision Assistance for complex systems (SFERCA company)
- Digesting programs design (Prolabo company)
- Navigating helping in a Documents Base (Chemdata company)
- CAD-CAM design helping (Dassault Systems / Catia V5)
 - ◆ Activity oriented assistance (first work)
 - ◆ Catia companion [TEBR] (last work)
- Current applications
 - ◆ Knowledge management [TEBR] (Antecim company)
 - ◆ Intelligent Tutoring Systems [TEBR] (Elycée company, National Institut of Research in Pedagogy)

Alain Mille

42

Discussion

- Open questions:
 - ◆ How to integrate really continuous learning processes in a CBR cycle
 - ◆ What's about the software engineering of Traced Experience Base Reasoning
 - Trace theory
 - Specific tools to modelize the way to trace activities when mediated by computers (with additional questions about confidentiality and security)
 - Specific tools to find past episodes in real time

Alain Mille

43

Thank you for your attention !

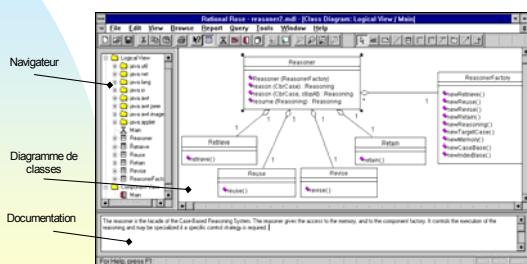
- Time for discussion ?

Alain Mille

44

Utilisation de CBR*Tools

- 1/ manipulation des classes java via éditeur
- 2/ atelier de manipulation directe des modèles UML



Alain Mille

45

CBR and TEBR applications

- CBR
 - ◆ Diagnosis
 - ◆ Planning
 - ◆ Designing
 - ◆ Help Desk
- TEBR
 - ◆ Personalized Help
 - ◆ Knowledge Management Facilitating
 - ◆ Collaborative work
 - ⇒ Human learning

Alain Mille

46

PROLABO / Programmation de minéralisateur micro-ondes

- Un programme de minéralisation est une sorte de recette (plan)
- Plusieurs cas sources et/ou un cas prototype sont utilisés pour créer un cas source à réutiliser
- L'adaptation est guidée par les différences structurelles relevées avec le cas cible
- Trois niveaux d'adaptation :
 - ◆ Le niveau plan.
 - ◆ Le niveau étape.
 - ◆ Le niveau valeur.

Alain Mille

47

Winxpert - Aliment special pour poudeuses

Fichier Edition Utilisateur Vue Applications Configuration ?

Description du cas

Référence: Essai Mas 27/01/95 Commentaires N. m. Aliment special pour

	Etape 1	Etape 2	Etape 3	Etape 4	Etape 5	Etape 6	Etape 7
Réactif	H2SO4	HNO3	<Aucun>		HClO4		HCl
Vitesse	7				5		6
Volume [ml]	2				2		5
Puiss.				15	20	30	30
Temp				6	14	18	5
Mise à sec	non			non	non	oui	non
Ote Sel(g)							

Une valeur

Un programme de digestion

une étape

Generaux Operatoire Matrice Protocole

Enregistrer Abandon Aide

estimation: 99.47

Winxpert - Aliment special pour poudeuses

Fichier Edition Utilisateur Vue Applications Configuration ?

Recherche bases de cas

Créa Référence: ref14 Nom: fourrage sec

	Etape 1	Etape 2	Etape 3	Etape 4	Etape 5	Etape 6
Réactif	HNO3		H2SO4	HClO4		HCl
Vitesse	8		7	5		6
Volume (ml)	20		3	4		5
Puissance (%)	15	25		30	30	30
Temps (min)	10	10				
Mise à sec	non	non	non			

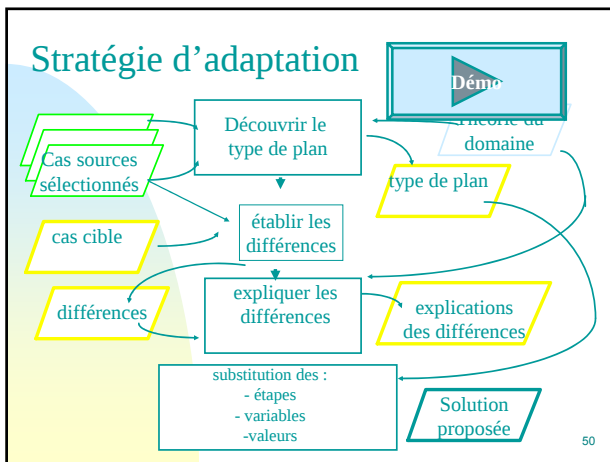
Plusieurs cas sources

un cas prototype

General Cas proposé Cas1/Cas2/Cas3/Cas4/Cas5 Proto/

Estimation: 99.47

Imprimer Retenir le cas pour adaptation Fermer Abandon Aide Accepter Adapter



RADIX

- Modélisation de l'utilisation d'un explorateur et des tâches de navigation et de recherche d'information
- Application à la recherche d'information en internet et intranet, en réutilisant l'expérience personnelle
- Partenaire industriel : Data Storage Systems by Chemdata

Alain Mille

51

Radix : les modèles

- **Modèle d'utilisation** : tout événement « faisant sens » dans le cadre de l'application (explorateur) (lien distant, lien local, retour, avance, signet, etc.)
- **Modèle tâche** : une interprétation des actions :
- **Session unitaire (SU)** : du début à la fin d'un épisode de recherche d'information
- **Tentative unitaire (TU)** : une recherche cohérente autour d'un sous-but particulier
- **Recherche Unitaire (RU)** : un triplet état-transition-état passant d'une « page » à une autre « page » de la recherche.
- **Vocabulaire utile (VU)** : les termes « gagnants » pour décrire une page « utile » (portée RU, TU ou SU)

Alain Mille

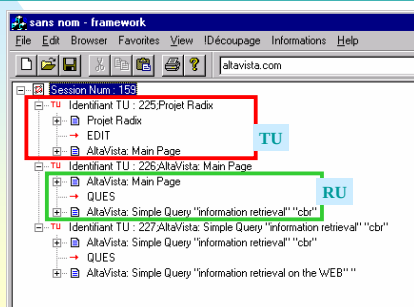
52

Illustration des modèles de Radix La session unitaire

Alain Mille

53

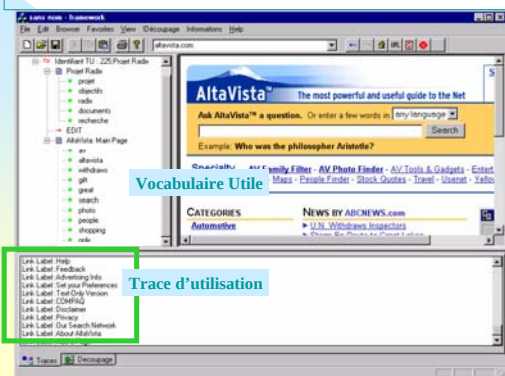
Illustration des modèles de Radix TU et RU



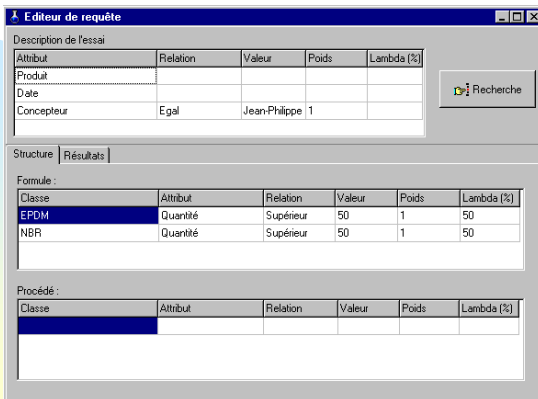
Alain Mille

54

Radix : connecter le modèle d'utilisation et le modèle de tâche

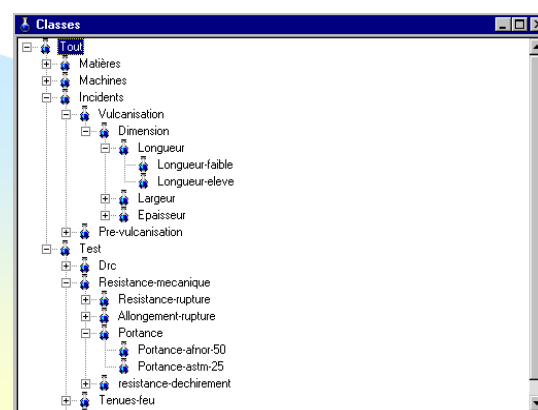


55



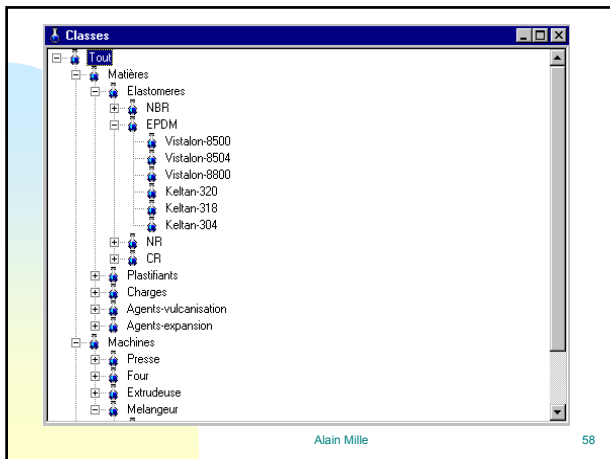
Alain Mille

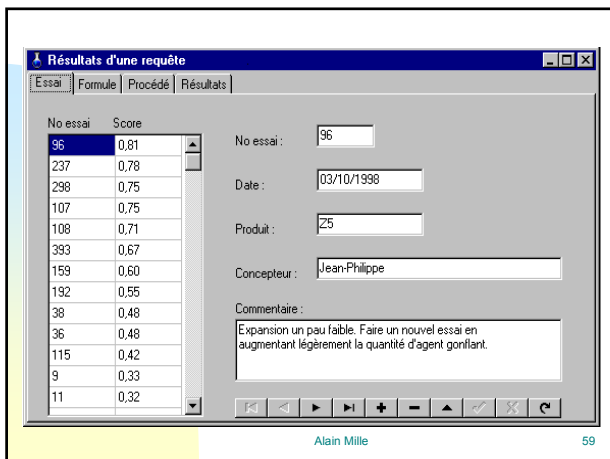
56



Alain Mille

57





JColibri – Historique

- Développé par le groupe GAIA (Group for Artificial Intelligence Applications, Univ. Complutense, Madrid)
- Origines :
 - ◆ Colibri [Bélen Diaz-Agudo, 2002]
 - ◆ CBRonto
 - ◆ LOOM (Logiques de description) + LISP
- JColibri
 - ◆ JAVA
 - ◆ Philosophie opensource
 - ◆ V1.0 beta

JColibri – Présentation

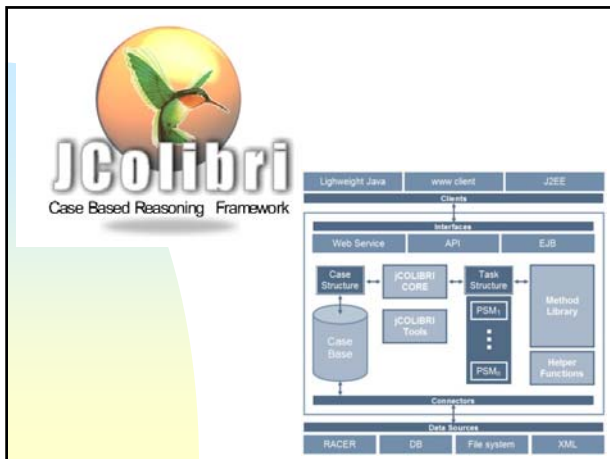
- Framework de Prototypage rapide d'applications de RàPC
- Approche tâches-méthodes. Deux types de méthodes :
 - ◆ Décomposition
 - ◆ Résolution
- Connecteurs :
 - ◆ Fichiers texte
 - ◆ Bases de données
 - ◆ RACER
 - ◆ ...
- Génération d'applications autonomes

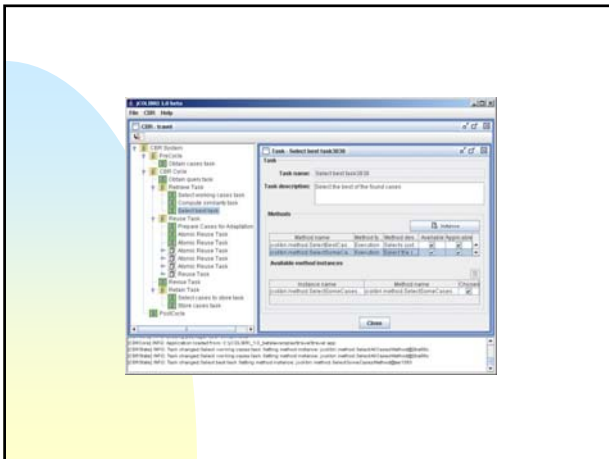
JColibri – Eléments clés

- Possibilités de configurer :
 - ◆ Structure du cas
 - ◆ Connecteurs
 - ◆ Tâches/Méthodes
 - ◆ Types de données personnalisés
 - ◆ Mesures de similarité
- Pas de base de connaissance !

JColibri – Possibilités d'extension

- Développement de tâches/méthodes personnalisées
 - ◆ Description en XML (automatisé par JColibri)
 - ◆ Implantation en Java
- Développement de mesures de similarité
- Utilisation des composants supplémentaires fournis avec JColibri ou par d'autres membres de la communauté :
 - ◆ CRN
 - ◆ Racer
 - ◆ Composants personnalisés

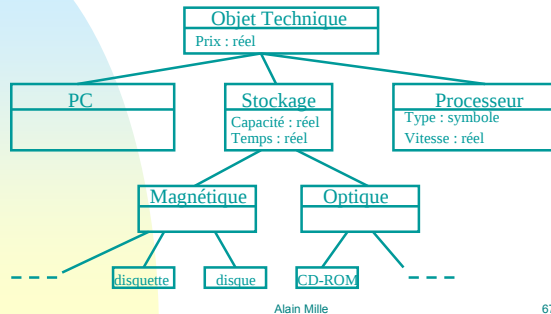




Aspects de la similarité

- K-plus proches voisins.
- Agrégation.
- Recherche selon point de vue.
- Prise en compte de la dynamique d'une séquence.
- Approches inductives.

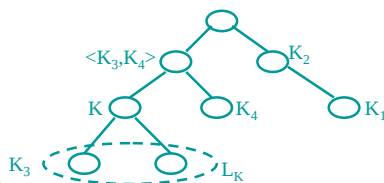
Représentation objet et Similarités



Alain Mille

67

Calcul de similarités-1



◆ Similarité intra-classes

- ⇒ propriétés communes entre deux objets.
- ⇒ similarité sur la classe commune la plus spécifique.
- ⇒ $Sim_{intra} = F(sim_{A1}(q.A_1, c.A_1), \dots, sim_{An}(q.A_n, c.A_n))$,

Alain Mille

68

Calcul de similarités-2

OBJETS CONCRETS

- $Sim(q, c) = Sim_{intra}(q, c) \cdot Sim_{inter}(class(q), class(c))$
- Similarité inter-classes
 - ◆ $Sim_{inter}(K, K_1) \leq Sim_{inter}(K, K_2)$ IF $\langle K, K_1 \rangle > \langle K, K_2 \rangle$
 - ◆ Associer une similarité S_i à chaque nœud
 - ⇒ X, Y dans L_{K_i} , $Sim_{inter}(X, Y) = S_i$
 - ◆ $Sim_{inter}(K_1, K_2) =$
 - ⇒ 1 si $K_1 = K_2$
 - ⇒ $S_{\langle K_1, K_2 \rangle}$ sinon

Alain Mille

69

Calcul de similarités-3

- Objet abstrait et requêtes
 - ◆ $\text{Sim}_{\text{inter}}(Q, C) = \max \{ \text{Sim}_{\text{inter}}(Q, C') \mid C' \text{ dans } L_c \}$
 - ☞ 1 si $Q < C$
 - ☞ $S_{<Q, C>}$ sinon
- Objets abstraits
 - ◆ $\text{Sim}_{\text{inter}}(Q, C) = \max \{ \text{Sim}_{\text{inter}}(Q, C') \mid Q' \text{ dans } L_q, C' \text{ dans } L_c \}$
 - ☞ 1 is $Q < C$ ou $C < Q$
 - ☞ $S_{<Q, C>}$ sinon

Alain Mille

70

Illustration simple sur un cas de vente d'automobiles d'occasion

- La base de cas est constituée d'épisodes de vente
- Le problème est décrit par les descripteurs du véhicule
- La solution est le prix de vente réellement négocié

Alain Mille

71

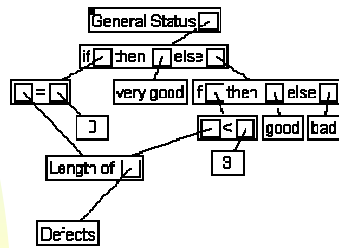
Descripteurs de cas

Field Name	Field Value	Field Type
Age du véhicule	n/a	Real
cas solution	n/a	Case
Etat général	Mauvais	Symbol (Formula)
Kilométrage	277000	Integer
Liste de défauts	(Corrosion superficielle, R)	List of Symbol
Marque	Peugeot	Symbol (Formula)
Modèle	206	Symbol
Prix de vente	10000.00	Real
Type de véhicule	Break	Symbol

Alain Mille

72

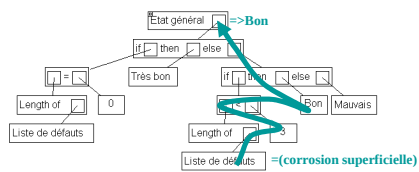
Élaboration / Ontologie du domaine



Alain Mille

73

Elaboration / Règle



Alain Mille

74

Élaborer : résumé

- Affectation des descripteurs au nouveau cas.
- Construire des descripteurs possédant une sémantique liée au problème.
- Anticiper au maximum l'adaptabilité des cas qui seront remémorés.

Alain Mille

75

Exemple : Élaborer dans ACCELERE

Assistance à la conception de caoutchouc

Trois types de tâches à assister :

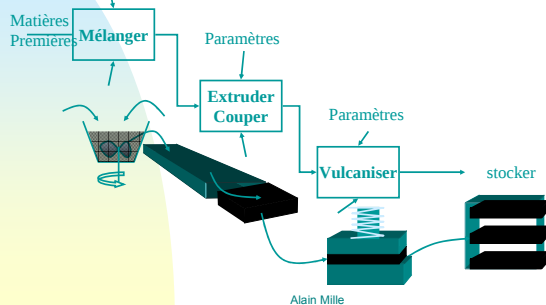
- **Synthèse** : trouver une structure permettant de satisfaire des spécifications
- **Analyse** : trouver le comportement résultant d'une structure particulière
- **Évaluation** : vérifier que le comportement est conforme à ce qui est attendu

Alain Mille

76

Le processus de production de caoutchouc

Paramètres de fabrication



77

Concevoir un nouveau produit

Synthèse
d'une structure
pour atteindre
les spécifications

TESTS
Ana

Un essai = un cas

Plusieurs centaines d'essais pour un produit
Plusieurs mois de mise au point

Alain Mille

78

Aide à l'élaboration..

Élaboration d'indices

État désiré:
 Force Choc E=50J INTEREP selon EN 1621-1 = [Très bas, Bas]
 Dureté Shore 00 intérieur = [Moyen, Élevé, Très élevé]

État le plus proche atteint avec:
 Force Choc E=50J INTEREP selon EN 1621-1 = Très bas
 Dureté Shore 00 intérieur = Moyen
Structure:
 NBR/PVC, Pcc = 100
Similarité: 91%

Alain Mille

80

Exploitation pour la recherche..

Essai	Similitude	Coef. Pcc	Volume In	Volume E	Volume T	Com
7031	0.78					
500	0.61					
1010	0.67					
595	0.67					
596	0.66					
594	0.66					
1009	0.66					
593	0.66					
597	0.64					
245	0.63					
522	0.42					
519	0.41					
90	0.4					
183	0.4					
468	0.39					
503	0.39					
501	0.39					
471	0.39					
504	0.39					

Mesure de similarité

- Se rappeler qu'elle doit être représentative de l'adaptabilité !
- Littérature très abondante sur le sujet :
 - ◆ la base (Tversky)
 - ⇒ $\text{Sim}(a,b) = f(A \cap B) / f(A \cup B)$
 - $A = \{\text{caractéristiques de } a\}$
 - $B = \{\text{caractéristiques de } b\}$
 - ◆ Biblio : B. Bouchon-Meunier, M. Rifqi and S. Bothorel, *Towards general measures of comparison of objects*, *Fuzzy Sets and Systems*, vol. 84, n.2, p. 143-153, 1996.

Alain Mille

82

Mesures de similarités

- Prendre en compte la structure de cas(Mignot)
- Mesures de comparaisons(Rifqi)
 - ◆ Mesure de similitudes
 - ◆ Mesures de dissimilarité
- Prendre en compte des historiques & des séquences (Mille, Jaczynski, Rougegrez)
- Etc.

Alain Mille

83

Adaptation générative

- trace de raisonnement = plan de résolution + justifications (+ alternatives + tentatives ayant échoué...)
- moteur de résolution complet = système de résolution de contraintes, planificateur, recherche dans un espace d'états, etc.

Alain Mille

84

Résolution de contraintes

- Cadre [HFI96]
- Notion de réduction de « dimensionnalité » fondée sur l'interchangeabilité et la résolution de contraintes.
- Représentation explicite des degrés de liberté pour l'adaptation :
 - ◆ 1) les contraintes liées aux anciens éléments de contexte ayant changé sont relâchées,
 - ◆ 2) on ajoute les contraintes liées aux nouveaux éléments de contexte.
 - ◆ 3) on résout le jeu réduit de contraintes.

[HFI96] : Kefeng Hua, Boi Faltings, and Ian Smith.
Cadre: case-based geometric design. *Artificial Intelligence in Engineering*, pages 171-183, 1996

Alain Mille

85

Planificateur

- On cherche un plan qui satisfasse aux mieux les buts à atteindre à partir de l'état initial (état final proche).
- Ce plan est généralisé (le moins possible) pour donner un état intermédiaire susceptible de conduire à l'état final recherché.
- A partir de cet état intermédiaire, on tente de terminer la planification.
- En cas d'échec, on remonte dans l'arbre des états pour généraliser un peu plus le plan...

Alain Mille

86

Autres approches de l'adaptation

- Par recherche en mémoire et **application de cas d'adaptation** : acquisition progressive de compétences d'adaptation.
- Par cycles itératifs de raisonnement à partir de cas : **décomposition hiérarchique de l'adaptation**
- Par l'étude **des similarités de rôle** des éléments dans le cas.
- **Selon un chemin de similarité**

Alain Mille

87

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

Alain Mille

88

Révision : l'exemple de CHEF*

- CHEF est un système de planification.
- Explication d'un échec par instanciation d'un arbre causal.
- Réparation par des connaissances générales.
- Échecs et succès guident l'insertion dans la mémoire (mémoire dynamique).

*CHEF est un exemple « historique »

Alain Mille

89

Maintenir une base de cas Qualité des cas-1

- Heuristiques (Kolodner)
 - ◆ Couvrir la tâche de raisonnement.
 - ◆ Couvrir les situations de succès et les situations d'échec.
 - ◆ Cas collectionnés d'une manière incrémentale.

Alain Mille

90

Maintenir une base de cas Qualité de cas-2

- Utilité
 - ◆ Par rapport à la performance.
 - ◆ Se débarrasser des connaissances inutiles.
- Compétence
 - ◆ Couverture d'un cas.
 - ◆ Accessibilité d'un problème.

Alain Mille

91

Maintenir une base de cas Modéliser la compétence-1 (Smyth)

- Cas essentiels : dont l'effacement réduit directement la compétence du système.
- Cas auxiliaires : la couverture qu'il fournit est subsumée par la couverture de l'un de ses cas accessibles.
- Cas ponts : leurs régions de couverture feront la liaison entre des régions qui sont couvertes indépendamment par d'autres cas.
- Cas de support : cas ponts en groupe.

Alain Mille

92

Maintenir une base de cas Modéliser la compétence-2



Compétence

★ Auxiliaire,
Support,
Pont,
Essentiel.

Alain Mille

93

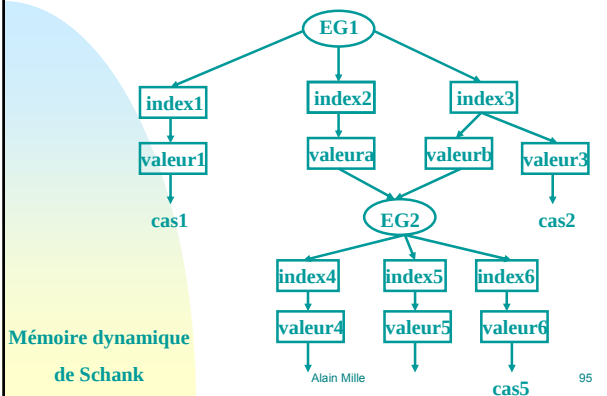
Organisation des cas

- Mémoire plate
 - ◆ Indexation superficielle
 - ◆ Partitionnement de la mémoire
 - ◆ Extraction parallèle
- Mémoire hiérarchique
 - ◆ Réseaux à traits partagés
 - ◆ Réseaux de discrimination

Alain Mille

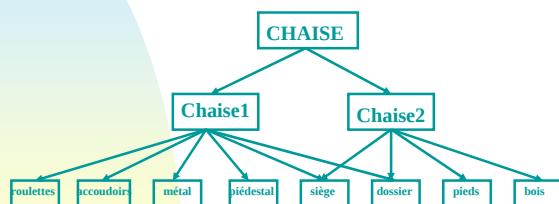
94

Réseaux de discrimination



Apprendre des connaissances-1

Exemple : Protos (cas + indexation)

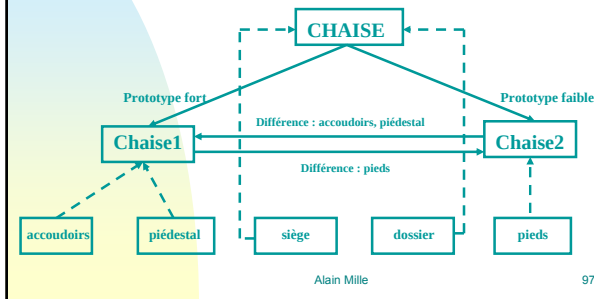


Alain Mille

96

Apprendre des connaissances-2

Exemple : Protos (cas + indexation)



Maintenance de la base de cas (Leake98)

- Stratégies
 - ◆ Collecte des données
 - périodique, conditionnel, Ad Hoc.
 - ◆ Intégration des données
 - On-line, Off-line.
- Activation de la maintenance
 - espace, temps, résultat de résolution.
- Étendue de la maintenance
 - Large, étroite.

Alain Mille

98

Approches connexes au RàPC Exemples, Instances & Cas

- Raisonnement fondé sur la mémoire
 - ◆ Pas de théorie sur le domaine
 - ◆ Aucune tâche d'induction ou d'abstraction
- Apprentissage à partir d'instances
 - ◆ Instance = attribut-valeur
 - ◆ IBL (Aha), C4.5, ID5R(Quinlan)
- Exemples typiques (exemplar)
 - ◆ Protos
- RàPC conversationnel (Aha)

Alain Mille

99

Intégration avec d'autres approches

- Exemple : Règles + cas
 - ◆ Mode d'intégration
 - ☞ Coopératif
 - ☞ Intégration des règles dans le RàPC
 - ◆ Creek (Aamodt), Cabata (Lenz)
- Mode coopératif
 - ◆ A qui donner la main ?
 - ☞ Degrés de confiance
 - ☞ Selon type de cas

Alain Mille

100

Exemples d'outils et application

- [L'outil Orange \(http://www.empolis.com/\)](http://www.empolis.com/)
- [Kaidara \(http://www.kaidara.com/\)](http://www.kaidara.com/)
- [CaseBank \(http://www.casebank.com/\)](http://www.casebank.com/)
- [L'environnement JColibri](#)
- [L'outil Remind](#)
- [L'outil CBR-tools](#)
- [Application Prolabo](#)
- [Application Interep](#)
- [Application Radix](#)

Alain Mille

101

Outil CBR*Tools

- Action AID, INRIA Sophia-Antipolis
- M. Jaczynski & B. Trousse
- Constat : Manque d'ouverture des outils existants (modification, ajout de composants difficile ou impossible)
- Nouveau type d'outil en RàPC : Plate-Forme a objets (*en Java*)
- Architecture - Points d'ouverture
- Modèles a objets - Explication en termes de patrons de conception
- Contact: trousse@sophia.inria.fr

Alain Mille

102
