

# Raisonnement à Partir de Cas

Introduction à l'attention des étudiants de Maîtrise Lyon1

Alain Mille

## Introduction

- Objectif du cours : fournir les définitions, les principes, les méthodes, les techniques et des exemples pour la mise en œuvre d'applications du RàPC.

Alain Mille

2

## Plan général du cours

- Racines historiques
- Principes de base du RàPC.
- Étude des différentes phases du cycle RàPC (Élaboration, Remémoration, Adaptation, Révision, Mémorisation).
- Exemples d'applications et d'outils.

Alain Mille

3

## Racines, Principes

- Minsky, un modèle de mémoire.
- Schanck, auteur de l'expression « Case-Based Reasoning ».
- Principes directeurs du Raisonnement à Partir de Cas

Alain Mille

4

## Minsky, le modèle de mémoire : principe

« Quand on rencontre une nouvelle situation (décrite comme un changement substantiel à un problème en cours), on sélectionne de la mémoire une structure appelée « cadre » (frame). Il s'agit d'une structure remémorée qui doit être adaptée pour correspondre à la réalité en changeant les détails nécessaires. »

Alain Mille

5

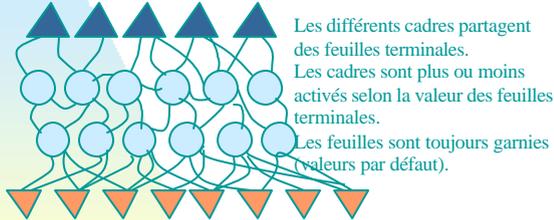
## Minsky, le modèle de mémoire : les cadres

- Une partie de l'information concerne son usage,
- une autre partie concerne ce qui peut arriver ensuite,
- et une autre partie concerne ce qu'il convient de faire en cas d'échec (quand ce qui devait arriver n'arrive pas...).

Alain Mille

6

## Minsky, le modèle de mémoire : illustration



Alain Mille

7

## Minsky, le modèle de mémoire : processus

- Les cadres sont des situations « idéales » regroupées en hiérarchie et sont reliés par les différences qui les séparent.
- Processus :
  - ◆ sélectionner un cadre,
  - ◆ tenter d'appliquer le cadre (faire le bilan des buts non atteints),
  - ◆ appliquer une technique d'adaptation-correction,
  - ◆ synthétiser l'expérience pour l'ajouter à la bibliothèque de techniques de correction.

Alain Mille

8

## Schank et le modèle de mémoire dynamique

- « Comprendre c'est expliquer ».
- Problématique de la compréhension des textes en langage naturel.
- Utilisation de scripts pour expliquer des situations.
- Utiliser l'expérience concrète dans la construction des scripts.

Alain Mille

9

## Des scripts à la mémoire dynamique



Alain Mille

10

## Processus de raisonnement

- Dans une mémoire d'expériences, organisée en hiérarchie de généralisation,
- on recherche ce qui est le plus près du problème courant,
- on réutilise le script trouvé en le spécialisant dans le contexte courant,
- on réorganise la mémoire pour y intégrer le nouvel épisode.

Alain Mille

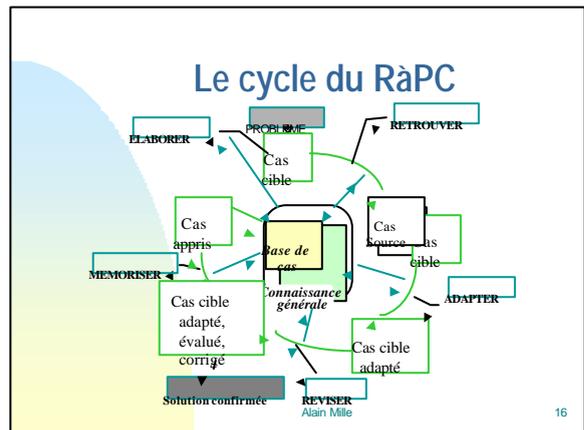
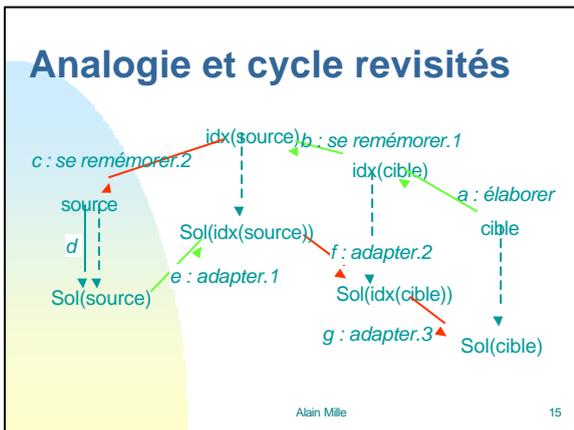
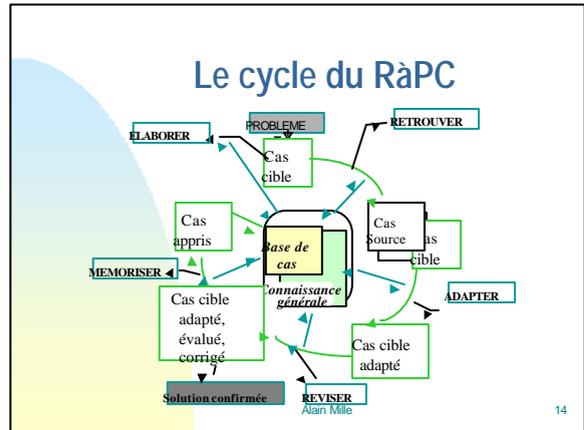
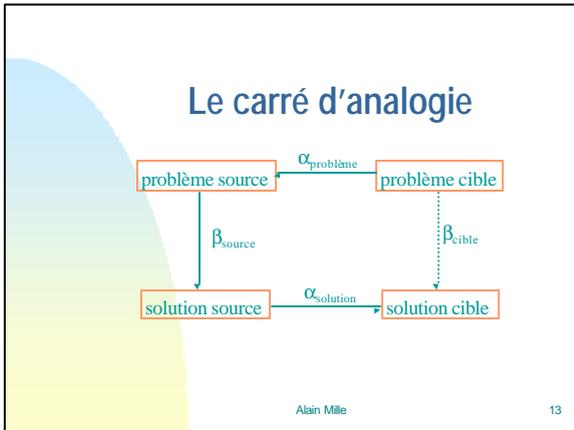
11

## Principes du RàPC

- Le carré d'analogie
- Le cycle du RàPC
- L'analogie et le cycle revisités...

Alain Mille

12



- ### Élaborer
- Rappel : on cherche 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
- Alain Mille 17

- ### 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 18



## Retrouver

- 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.

Alain Mille 25

## 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)

Alain Mille 26

## 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.

Alain Mille 27

## Représentation objet et Similarités

Alain Mille 28

## Calcul de similarités-1

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

Alain Mille 29

## 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  $S_j$  à chaque nœud
    - +  $X, Y$  dans  $L_{K_j}$ ,  $Sim_{inter}(X, Y) = S_j$
  - ◆  $Sim_{inter}(K_1, K_2) =$ 
    - + 1 si  $K_1 = K_2$
    - +  $S_{\langle K_1, K_2 \rangle}$  sinon

Alain Mille 30

## Calcul de similarités-3

- Objet abstrait et requêtes
  - ◆  $Sim_{inter}(Q,C) = \max \{Sim_{inter}(Q,C') \mid C' \text{ dans } Lc\}$ 
    - + 1 si  $Q < C$
    - +  $S_{<Q,C>}$  sinon
- Objets abstraits
  - ◆  $Sim_{inter}(Q,C) = \max \{Sim_{inter}(Q,C') \mid Q' \text{ dans } Lq, C' \text{ dans } Lc\}$ 
    - + 1 si  $Q < C$  ou  $C < Q$
    - +  $S_{<Q,C>}$  sinon

Alain Mille

31

## Adapter : la problématique

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

Alain Mille

32

## Exemple : la configuration d'un ordinateur multimédia\*

- L'utilisateur spécifie les applications qu'il souhaite exploiter (traitement de texte, musique, programmation, jeux).
- Chaque logiciel est noté selon l'importance accordée par l'utilisateur.
- L'objectif est d'élaborer la configuration idéale supportant les logiciels demandés en fonction de leur importance.
- La solution est représentée selon une structure « objet » d'un PC et de ses composants.

\* exemple tiré de [BerWil98]

Alain Mille

33

## Adapter : deux approches

- Adaptation **générationnelle** : on a toutes les connaissances pour résoudre le problème à partir de zéro.
- Adaptation **transformationnelle** : on n'a pas toutes les connaissances pour résoudre le problème à partir de zéro.

Alain Mille

34

## Adaptation générative

- Le cas retrouvé retrace le « raisonnement » ayant mené à la solution.
- On substitue les éléments de contexte du raisonnement retrouvé par les éléments différents du contexte du cas nouveau.
- On « rejoue » le raisonnement dans ce nouveau contexte

Alain Mille

35

## Exemple / configuration

Nouveau cas

Jeux = 0;  
Musique = 10;  
TdT = 5;  
Prog = 5;  
(Puissance = 10)

Cas retrouvé

Jeux = 10;  
Musique = 0;  
TdT = 5;  
Prog = 5;  
(Puissance = 10)

trace du raisonnement...

- 1) Sélectionner carte-mère (>ASUS)
- 2) Sélectionner CPU (>pentium 200)
- 3) Sélectionner carte graphique (>Matrox)
- 4) Sélectionner le « joystick » (>JK485)
- 5) Sélectionner le CD-ROM (>Sony 10x)

Alain Mille

36

### Exemple / configuration

<b>Nouveau cas</b> - Jeux = 0; - Musique = 10; - TdT = 5; - Prog = 5; (Puissance = 10)	<b>Cas retrouvé</b> - Jeux = 10; - Musique = 0; - TdT = 5; - Prog = 5; (Puissance = 10)
---	--

*trace du raisonnement...*

- 1) Sélectionner carte-mère (>ASUS)
- 2) Sélectionner CPU (>pentium 200)
- 3) Sélectionner carte graphique (>Matrox)
- 4) Sélectionner le « joystick » (>JK485)
- 5) Sélectionner le CD-ROM (>Sony 10x)

Alain Mille 37

### Exemple / configuration

<b>Nouveau cas</b> Jeux = 0; Musique = 10; TdT = 5; Prog = 5; (Puissance = 10)	<b>Cas retrouvé</b> Jeux = 10; Musique = 0; TdT = 5; Prog = 5; (Puissance = 10)
---	--

*trace du raisonnement...*

- 1) Sélectionner carte-mère (>ASUS)
- 2) Sélectionner CPU (>pentium 200)
- 3) Sélectionner carte graphique (>Matrox)
- 4) Sélectionner le « joystick » (>JK485)
- 5) Sélectionner le CD-ROM (>Sony 10x)

Alain Mille 38

### Exemple / configuration

<b>Nouveau cas</b> - Jeux = 0; - Musique = 10; - TdT = 5; - Prog = 5; (Puissance = 10)	<b>Cas retrouvé</b> - Jeux = 10; - Musique = 0; - TdT = 5; - Prog = 5; (Puissance = 10)
---	--

*trace du raisonnement...*

- 1) Sélectionner carte-mère (>ASUS)
- 2) Sélectionner CPU (>pentium 200)
- 3) Sélectionner le CD-ROM (>Sony 10x)
- 4) Sélectionner l'adaptateur graphique (>ES3)
- 5) Sélectionner la carte son (>midi634)

Alain Mille 39

### 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 40

### Résolution de contraintes

- Cadre [HF196]
- 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.

[HF196] : Kefeng Hua, Boi Faltings, and Ian Smith.  
Cadre: case-based geometric design. Artificial Intelligence in Engineering, pages 171--183, 1996

Alain Mille 41

### 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 42

### Adaptation transformationnelle

- Des éléments de la solution du cas retrouvé sont :
  - modifiés,
  - supprimés,
  - ajoutés, selon
- des écarts de contexte observés entre cas source et cas cible, et grâce à
- un ensemble de règles d'adaptation.

Alain Mille 43

### Exemple / configuration

<b>Nouveau cas</b> Jeux = 0; Musique = 10; TdT = 5; Prog = 5; (Puissance = 10)	<b>Cas retrouvé</b> Jeux = 10; Musique = 0; TdT = 5; Prog = 5; (Puissance = 10)
---	--

**Solution**  
 Carte ASUS-3  
 Processeur pentium 250  
 Carte graphique Matrox G2  
 Joystick JK600  
 CD-Rom Sony 14X

Alain Mille 44

### Règles d'adaptation

- Si (source.jeu > 7) et (cible.jeu < 3) alors solution.supprimer(Joystick)
- Si (source.musique < 3) et (cible.musique > 7) alors solution.ajouter(carte\_son\_haut\_de\_gamme)
- Si (source.puissance < 3) et (cible.puissance > 7) alors solution.modifier(processeur.vitesse,delta\_puissance,+)

Alain Mille 45

### Exemple / configuration

<b>Nouveau cas</b> Jeux = 0; Musique = 10; TdT = 5; Prog = 5; (Puissance = 10)	<b>Cas retrouvé</b> Jeux = 10; Musique = 0; TdT = 5; Prog = 5; (Puissance = 10)
---	--

**Solution**  
 Carte ASUS-3  
 Processeur pentium 250

**Solution**  
 Carte ASUS-3  
 Processeur pentium 250  
 Carte graphique Matrox G2  
 Joystick JK600  
 CD-Rom Sony 14X

Alain Mille 46

### Exemple / configuration

<b>Nouveau cas</b> - Jeux = 0; - Musique = 10; - TdT = 5; - Prog = 5; (Puissance = 10)	<b>Cas retrouvé</b> - Jeux = 10; - Musique = 0; - TdT = 5; - Prog = 5; (Puissance = 10)
---	--

**Solution**  
 Carte ASUS-3  
 Processeur pentium 250  
 Carte graphique S3

**Solution**  
 Carte ASUS-3  
 Processeur pentium 250  
 Carte graphique Matrox G2  
 Joystick JK600  
 CD-Rom Sony 14X

Alain Mille 47

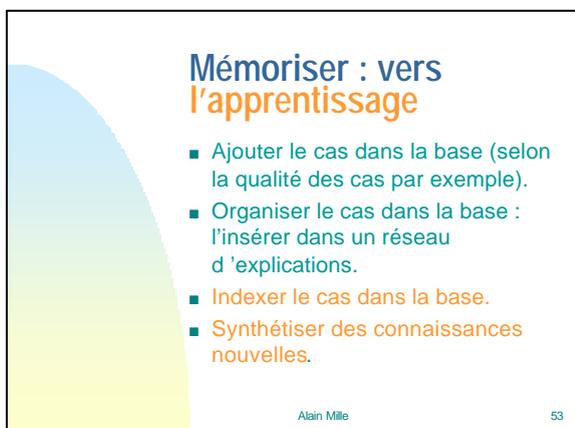
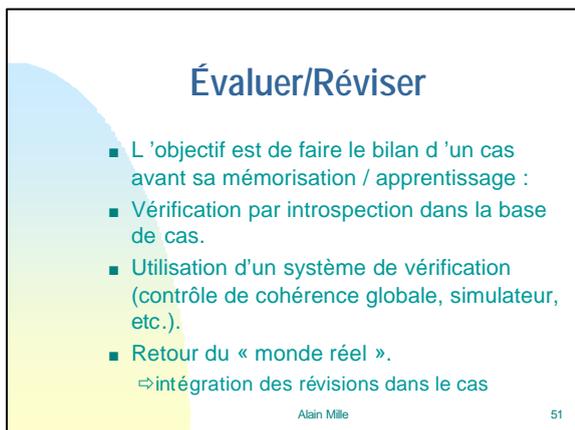
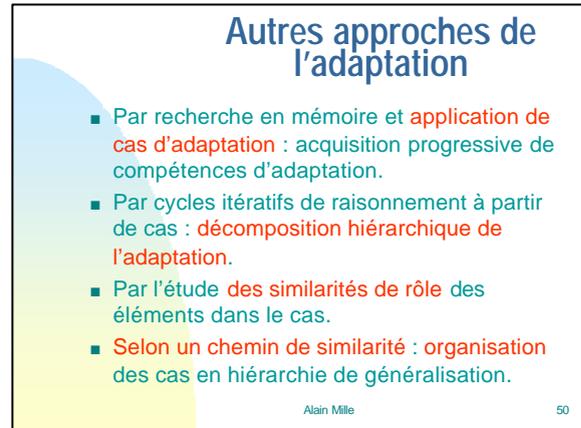
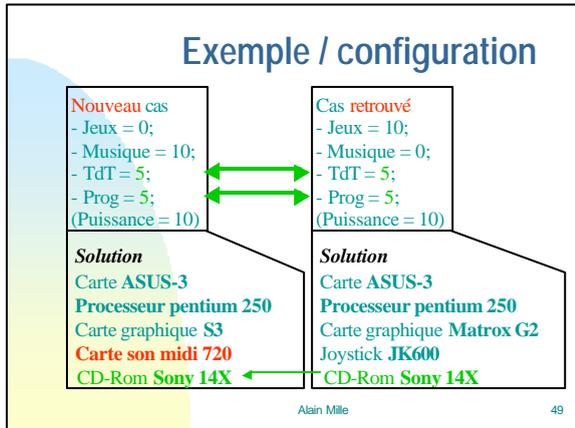
### Exemple / configuration

<b>Nouveau cas</b> - Jeux = 0; - Musique = 10; - TdT = 5; - Prog = 5; (Puissance = 10)	<b>Cas retrouvé</b> - Jeux = 10; - Musique = 0; - TdT = 5; - Prog = 5; (Puissance = 10)
---	--

**Solution**  
 Carte ASUS-3  
 Processeur pentium 250  
 Carte graphique S3  
 Carte son midi 720

**Solution**  
 Carte ASUS-3  
 Processeur pentium 250  
 Carte graphique Matrox G2  
 Joystick JK600  
 CD-Rom Sony 14X

Alain Mille 48



### 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 55

### 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 56

### Modéliser la compétence-2

Compétence

- ☆ Auxiliaire,
- ◇ Support,
- ☀ Pont,
- Essentiel.

Alain Mille 57

### Organisation des cas

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

Alain Mille 58

### Réseaux de discrimination

Mémoire dynamique de Schank

Alain Mille 59

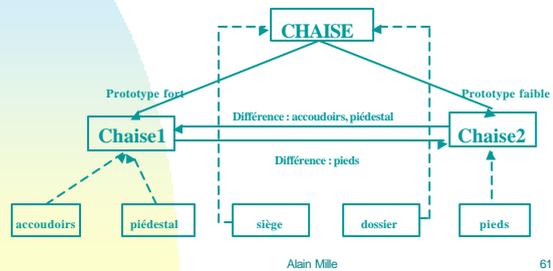
### Apprendre des connaissances-1

Exemple : Protos (cas + indexation)

Alain Mille 60

## Apprendre des connaissances-2

Exemple : Protos (cas + indexation)



Alain Mille

61

## 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

62

## 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

63

## 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

64

## Exemples d'outils et application

- [L'outil CBR-Works](#)
- [L'outil Remind](#)
- [L'outil CBR-tools](#)
- [Application Prolabo](#)
- [Application Interep](#)
- [Application Radix](#)
- [Application Broadway](#)

Alain Mille

65

## 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

66

### Utilisation de CBR\*Tools

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

Alain Mille 67

### 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 structurales relevées avec le cas cible
- Trois niveaux d'adaptation :
  - ◆ Le niveau plan.
  - ◆ Le niveau étape.
  - ◆ Le niveau valeur.

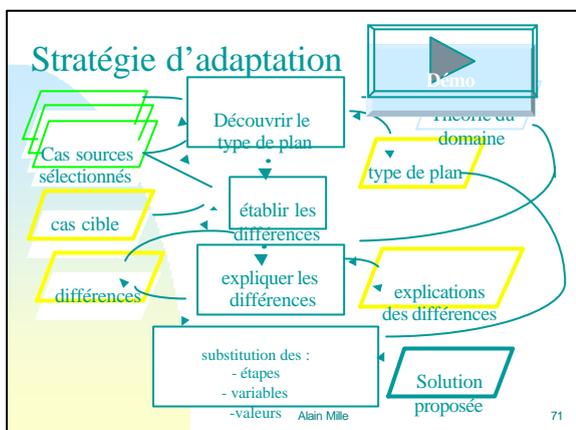
Alain Mille 68

Réactif	Etape 1	Etape 2	Etape 3	Etape 4	Etape 5	Etape 6	Etape 7	Etape 8
H2SO4	HNO3	<Aucun>			HClO4			HCl
Vitesse	7				5			6
Volume					2			5
Puissance				15	20	30		30
Temps (min)			5	6	14	18		5
Mise à sec	non			non	non	oui		non
Qte Sel(g)								

Alain Mille 69

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

Alain Mille 70



### 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 72

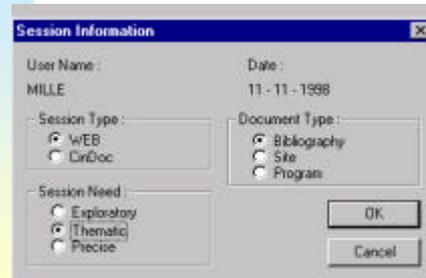
## Radix : les modèles

- **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

73

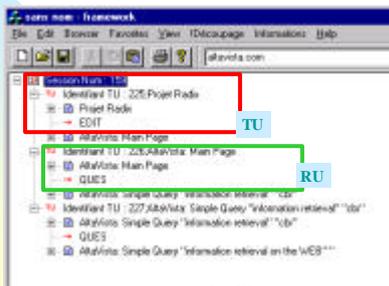
## Illustration des modèles de Radix La session unitaire



Alain Mille

74

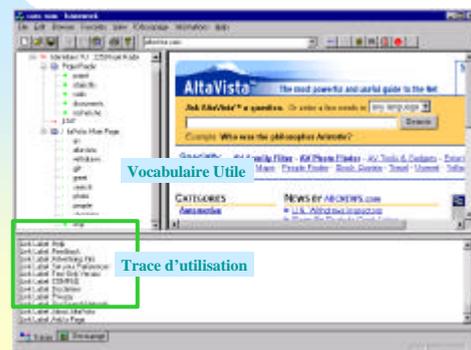
## Illustration des modèles de Radix TU et RU



Alain Mille

75

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



76

Editeur de requête

Description de l'essai				
Attribut	Relation	Valeur	Poids	Lambda (%)
Produit				
Date				
Concepteur	Egal	Jean-Philippe	1	

Recherche

---

Structure Résultats

Formule :					
Classe	Attribut	Relation	Valeur	Poids	Lambda (%)
EPDM	Quantité	Supérieur	50	1	50
NBR	Quantité	Supérieur	50	1	50

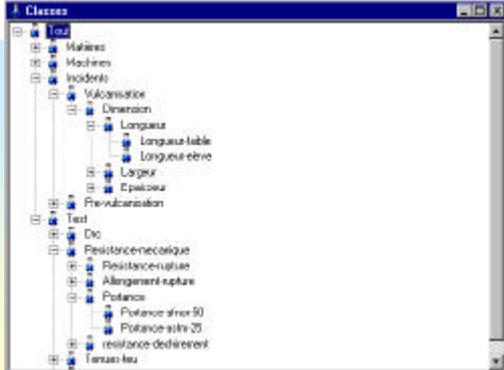
---

Procédés

Classe	Attribut	Relation	Valeur	Poids	Lambda (%)

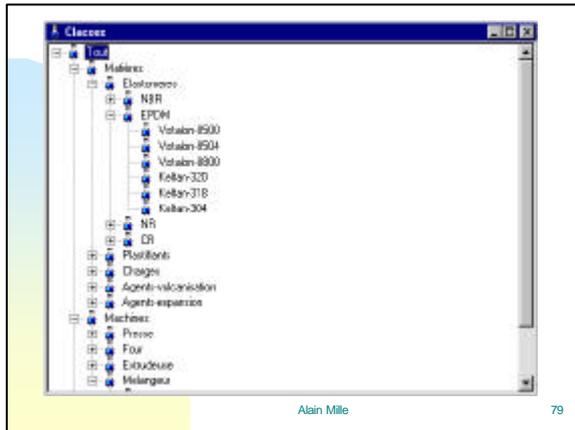
Alain Mille

77



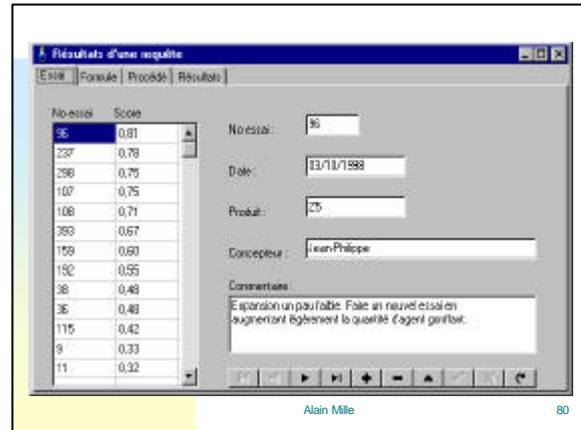
Alain Mille

78



Alain Mille

79



Alain Mille

80