



HAL
open science

Méthode de suppression de cas pour une maintenance de base de cas.

Mohamed-Karim Haouchine, Brigitte Chebel-Morello, Noureddine Zerhouni

► **To cite this version:**

Mohamed-Karim Haouchine, Brigitte Chebel-Morello, Noureddine Zerhouni. Méthode de suppression de cas pour une maintenance de base de cas.. 14ème Atelier de Raisonnement à Partir de Cas, RaPC'06., Mar 2006, Besançon, France. pp.39-50. hal-00339349

HAL Id: hal-00339349

<https://hal.science/hal-00339349>

Submitted on 18 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Méthode de Suppression de Cas pour une Maintenance de Base de Cas

Karim Mohamed Haouchine, Brigitte Chebel-Morello, et Nouredine Zerhouni

Laboratoire d'Automatique de Besançon
LAB – UMR CNRS 6596
24 rue Alain Savary – 25000 Besançon
{karim.haouchine, bmorello, zerhouni}@ens2m.fr

Résumé : Nous vous présentons un début de travail, concernant la maintenance de la base de cas. Après un état de l'art des travaux fait dans le domaine, nous sommes amenés à faire deux propositions, que nous n'avons pas encore exploitées. Une solution sur la suppression de cas et une autre en utilisant les outils d'apprentissage automatique, sachant que nous pouvons modéliser notre base de cas sous forme d'exemples.

La proposition se fait sur deux niveaux, sur des travaux d'apprentissage automatique et sur l'élaboration d'une nouvelle mesure.

Mots-clés : raisonnement à partir de cas, maintenance de la base de cas, politiques et stratégies de maintenance de la base de cas, apprentissage automatique, systèmes à base de connaissance.

1. Introduction

Tout système informatique arrivé à maturité, comme les systèmes traditionnels, les Systèmes à Base de Connaissances (SBC) et en particulier les systèmes de raisonnement à partir de cas (RàPC) nécessitent en phase de fonctionnement d'être maintenu, ne serait ce que mis à jour. Certes, il existe des facilités pour développer les SBC, mais ces systèmes ne rencontrent pas une adhésion générale, car ils se dégradent dans le temps, et il est essentiel de faire appel à des tâches permettant d'évaluer et de maintenir la qualité de ces systèmes. Ces tâches dites de maintenance sont essentielles et ne sont pas triviales à mettre en place. La maintenance ne se réduit pas à l'addition d'un nouvel élément dans la base de connaissances, comme cela s'est supposé dans un premier temps. Coenen dans [Coenen et al., 1999] souligne l'importance d'intégrer la phase de maintenance dans le cycle de vie des SBC. Il propose une méthodologie composée d'un certain nombre d'étapes qui inclue premièrement, l'identification des champs de la base de connaissances demandant à être modifier, puis dans une deuxième phase la détermination de la nature des actions de maintenance à mener (soit des erreurs subsistant dans la base de connaissances ou bien l'obtention de résultats erronés ou encore l'absence de résultats), et enfin une dernière phase de test des effets relatifs aux modifications opérées. Maints travaux dans la dernière décennie abordent les questions liées aux aspects de qualité et de sûreté dans les SBC. Les recherches concernant la Validation et Vérification (V&V) se sont alliées aux recherches sur les SBC [Nick et al., 2000] pour tenir compte conjointement des aspects de qualité et de sûreté des SBC. Ces travaux ont menés à la définition d'un cycle de vie de SBC, conçu pour intégrer des méthodes formelles et déboucher sur des méthodes de certification de logiciel. La notion de *raffinement* d'une base de connaissance a été définie comme l'amélioration ou la modification de la base de connaissance incohérente dans l'objectif de reconstituer sa concordance. Des études similaires

concernant plus spécifiquement les systèmes de RàPC sont faites. On retrouve les orientations vers la modélisation du cycle de vie du système de RàPC intégrant les phases liées à la maintenance et les méthodes travaillant sur la qualité du système qui est amené dans sa définition même à évoluer. On note que la qualité d'un système de RàPC est liée à la définition et à la représentation d'un cas, l'organisation de la **Base de Cas (BC)**, les diverses indexations utilisées et la définition de *bonnes* mesures de similarités pour la recherche des cas et le lien recherche adaptation des cas. Leake [Leake et al., 1998] constate que les différentes approches de maintenance de cas, peuvent se trouver aussi bien dans les recherches de RàPC que de système expert, sans compter les travaux d'apprentissage automatique, et les travaux sur les connaissances.

Dans le premier paragraphe, une étude de la maintenance des systèmes de RàPC est faite, en partant des travaux sur les différentes sources de connaissance en donnant un inventaire rapide des recherches dans le domaine en apprentissage. Plusieurs auteurs spécifient que cette maintenance peut se réduire à une maintenance de la BC du système. Par conséquent, un état de l'art sera présenté sur ce sujet et abordera au deuxième paragraphe les politiques et les stratégies développées pour maintenir une BC. On s'intéressera à l'optimisation de la BC et plus particulièrement aux stratégies d'ajout et de suppression de cas ainsi que les critères associés. Puis enfin, nous présenterons deux méthodes de suppression de cas à creuser : la première utilise des algorithmes d'apprentissage automatique, et la deuxième propose une mesure de compétence (*MC*), permettant d'optimiser la dimension de la BC.

1.1. La maintenance d'un système de RàPC

La maintenance met en application des politiques de mise à jour des représentations de la BC et s'intéresse à sa réorganisation afin de faciliter le raisonnement futur répondant à un ensemble d'objectifs de performance [Simon et al., 2001]. Richter dans [Richter, 1995], [Richter, 1998] a proposé la notion de source de données représentant des connaissances pertinentes du système dans un système de RàPC. Cette maintenance consiste à développer des techniques afin de contrôler et réagir face aux changements de ces différentes sources de connaissances. Ces sources sont :

- *La source de vocabulaire* : contient toutes les informations sur les définitions et les structures utilisées dans le système de RàPC
- *La source des mesures de similarité* : contient les mesures nécessaires pour la recherche des cas
- *La source d'adaptation* : contient les règles de transformation de la solution
- *La source de la base de cas* : représente le contenu et l'organisation de la BC

Les trois premières sources de connaissances, sont élaborées avant que le système tourne, alors que la BC est remise à jour normalement dynamiquement. Selon Richter, chaque source peut porter presque toute la connaissance disponible, et les manipulations sur une source ont de petites répercussions sur les autres. A chaque source de connaissance on peut associer une maintenance. Thomas Gabel [Gabel, 2004] a travaillé sur l'apprentissage des mesures de similarité ; comme l'évaluation de poids associés aux descripteurs d'un cas ou l'acquisition de mesures de similarité grâce à des fonctions d'utilité issues d'une boucle de retour sur le traitement associées à des applications de RaPC. Thomas Roth-Berghofer, [Roth-Berghofer et al., 2001] part de la modélisation des quatre phases de RàPC d'Aamodt et Plaza [Aamodt et al., 1994] : La remémoration, l'adaptation, la validation et la mémorisation, pour les séparer en deux méta-phases en lien direct avec les sources de connaissances. La première concerne l'application et contient les phases de remémoration, d'adaptation et de réutilisation. Il est à noter que la première phase du cycle, la phase d'élaboration de cas est omise dans cette modélisation, mais a toute sa place. La deuxième méta phase concerne la maintenance et est

composée de trois phases : la mémorisation, la scrutation et la restauration. Iglezakis soutient dans [Iglezakis et al., 2000] qu'il ne peut y avoir de maintenance de système de RàPC sans le balayage de la BC. Que la BC est associée à la connaissance du système, que les opérations de maintenance ne sont déclenchées que grâce à elle. La BC a un rôle central ce qui explique que la majorité des travaux faits dans ce domaine, est essentiellement fondé sur la Maintenance de la Base de Cas (MBC). De plus les connaissances d'un système de RàPC sont liées aux cas puisque ceux-ci sont affectés par tout changement dans les sources de connaissances. La BC est la source de connaissance la plus sensible aux changements dans le système de RàPC, sa consultation est la plus appropriée pour déclencher les opérations de maintenance [Zehraoui, 2004].

1.2. La Maintenance de la Base de Cas

La MBC est la mise en œuvre des politiques permettant de réviser l'organisation et/ou le contenu (représentation, domaine d'application, contenu d'informations, ou l'implémentation) de la BC afin d'améliorer le raisonnement futur [Leake et al., 1998]. La MBC [Smyth, 1998a] est un ensemble de réalités différentes, telle que la suppression des cas incohérents, de la sélection de groupes de cas permettant d'éliminer la redondance et d'améliorer le pouvoir de raisonnement du système, les cas peuvent être réécrit afin de réparer les problèmes d'incohérences.

1.3. Maintenance de la base de cas en apprentissage automatique

[Anand et al., 1998] a proposé d'utiliser des techniques de fouille de données, dédié à l'acquisition de cas et la découverte de connaissances d'adaptation. [Hanney et al., 1996] ont présenté un algorithme d'apprentissage inductif de cas dans une BC. [Basak et al., 1998] ont utilisé des méthodes de logique floue et des réseaux de neurones, pour déterminer les caractéristiques importantes et éliminer les caractéristiques non pertinentes dans un ensemble de données. Les méthodes de sélection de variables permettent de réduire la dimension de la BC quand les caractéristiques non pertinentes sont éliminées [Wettscherk, 1995].

Il existe plusieurs stratégies de suppression de cas, les toutes premières parlent du problème d'utilité concernant l'apprentissage automatique. On parle du problème d'utilité lorsque la taille de la BC augmente, au point de devenir saturée et de réduire la performance du système de RàPC [Smyth et al., 1996].

Simon propose une méthode de maintenance composée de quatre étapes [Simon et al., 2001] :

- Une approche d'apprentissage automatique des poids associés aux caractéristiques, pour évaluer leur importance dans la BC.
- Partitionnement des cas pour identifier les différents concepts utilisant la connaissance sur les caractéristiques obtenues.
- Extraction des connaissances pour chaque concept en utilisant les arbres de décision flou.
- Finalement, une stratégie de sélection fondée sur les notions de recouvrement et d'atteignabilité.

2. Maintenance de la base de cas spécifique au RàPC

Il existe différentes politiques qui déterminent quand et comment un système de RàPC exécute la MBC. Ces politiques sont décrites par [Leake et al., 1998] et dépendent de :

- La manière de rassembler les données pertinentes pour la maintenance
- Le moment de déclenchement de la maintenance
- Les types d'opération de maintenance possibles

- La manière d'exécuter les opérations sélectionnées.

On peut diviser les approches proposées pour la MBC en deux catégories, la première traite le problème d'optimisation de la BC et la deuxième traite le problème du partitionnement de la BC (voir *figure 1*).

Une BC est de bonne qualité si elle permet au système de RàPC de résoudre le plus de problèmes possibles de manière correcte en un temps raisonnable [Zehraoui, 2004].

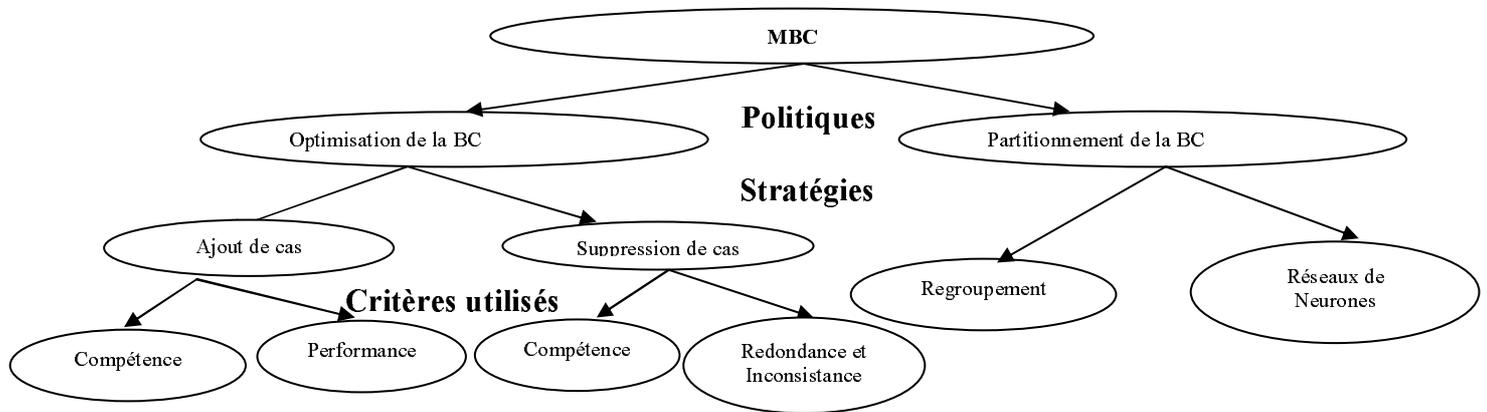


Figure 1. Schéma des différentes stratégies, types et critères utilisés dans la MBC

Nous décrivons deux notions centrales permettant d'évaluer la qualité de la BC.

- **La Compétence**, est mesurée par le nombre de problèmes différents pour lesquels le système apporte une bonne solution [Smyth et al., 1995].
- **La Performance**, d'un système est mesurée par le temps de réponse qui lui est nécessaire pour proposer une solution à un cas cible [Smyth et al., 1995]. Cette mesure est liée directement aux coûts d'adaptation et aux coûts de recherche.

Dans ce qui suit, nous allons aborder les deux politiques principales proposées dans la littérature.

2.1. Politique de partitionnement de la base de cas

Dans cette politique, il n'y a pas de suppression de cas mais un partitionnement de ceux-ci en plusieurs espaces de recherche ce qui permet de réduire ainsi le temps de recherche et de sélectionner de manière incrémentale les attributs qui sont riches en information et qui peuvent couvrir la structure de la BC entière [Yang et al., 2000].

On trouve également des stratégies utilisant les réseaux de neurones. Certains prennent en compte l'arrivée dynamique des données [Malek, 2000] tandis que d'autres utilisent des réseaux de neurones statiques [Roh et al., 2003].

2.2. Politique d'optimisation de la base de cas

Dans cette politique on réduit la taille de la BC ce qui diminue le temps de recherche. Il existe deux stratégies permettant cette optimisation, l'ajout et la suppression de cas. Dans chacune d'elle différentes méthodes sont développées et évalue la BC sous des aspects de *compétence* [Smyth et al., 1998b] ou de *performance* [Leake et al., 2000a].

Nous allons développer maintenant les deux stratégies décrites dans cette politique.

3. Stratégie d'ajout de cas

La stratégie d'ajout de cas consiste comme son nom l'indique à construire une BC réduite en partant d'une BC vierge par l'ajout successif de cas présentant un critère maximisé (critère de compétence ou de performance). De ce fait, il existe deux méthodes, chacune d'elles vise à maximiser l'un de ces critères.

3.1. Méthode maximisant le critère de compétence

Smyth et McKenna [Smyth et al., 1999b] ont proposé de construire une BC en utilisant un algorithme hybride, celui des plus proches voisins (CNN) en l'associant à une mesure qui sera appliquée pour chaque cas dans la BC originale. Cette mesure est la valeur du recouvrement relatif RC (Relative Coverage). Elle détermine la valeur de la contribution d'un cas à la compétence de la BC. Par conséquent, les cas sont rangés selon leurs valeurs de « RC » dans la BC complète. Ensuite l'algorithme CNN est appliqué pour construire une nouvelle BC réduite. Ceci permet de sélectionner les cas ayant une grande contribution au recouvrement de la BC. Q. Yang et J. Zhu [Yang et al., 2001] déterminent le recouvrement par une mesure de similarité et des coûts d'adaptation. Dans ce cas, le recouvrement d'un cas est considéré comme le voisinage du cas dans certaines limites d'adaptabilité.

3.2. Méthode maximisant le critère de performance

Leake et Wilson [Leake et al., 2000a] ont définis une mesure de performance relative RP (Relative Performance) inspirée des travaux de la mesure « RC » qui est basée sur le coût de l'adaptation de chaque cas de la BC. La valeur de la performance relative d'un cas « RP(c) » reflète sa contribution à la performance d'adaptation comparée à celle des autres cas. Avant d'ajouter un cas dans la BC réduite, sa contribution à la performance d'adaptation est estimée. Un ensemble de mesures incluant la mesure du bénéfice de performance produit par l'ajout de chaque cas est estimée. De ce fait, cette mesure utilisée pour guider l'ajout de cas en favorisant les cas ayant une grande valeur de « RP ».

Dans le même cadre une autre mesure a été développée dans [Leake et al., 2000b] concernant le bénéfice de la performance relative PB (relative Performance Benefit) qui est basée sur les mêmes notions que la mesure « RP », qui donne les mêmes performances c'est à dire le même taux de réduction de la taille de la BC.

Cependant, la technique RC-CNN fournit une réduction de la taille de la BC supérieure par rapport à celui des techniques RP-CNN et PB-CNN. Par contre, ces deux dernières donnent un résultat concernant le coût d'adaptation des cas dans la BC obtenue nettement meilleur par rapport à RC-CNN.

4. Stratégie de suppression de cas

A partir d'une BC, cette stratégie évalue les cas suivant un critère afin de pouvoir les supprimer et ramener la BC à un nombre de cas donné. Les critères d'évaluation tels que la compétence, la redondance et l'inconsistance ont été utilisés dans différentes méthodes.

4.1. Méthodes de suppression par balayage de la base de cas.

Dans ce cas, la BC est balayée entièrement dès que sa taille atteint un seuil prédéfini.

- *la suppression aléatoire*, consiste à supprimer aléatoirement un cas de la BC dès que sa taille dépasse un certain seuil [Markovitch et al., 1988]. Cette méthode sera un point de référence pour la comparaison avec les autres méthodes.

- *Ironically* [Minton, 1990], qui s'apparenterai à un facteur d'oubli de la connaissance la moins utilisée en calculant la *fréquence de remémoration* de chaque cas dans la BC.
- *Suppression en se basant sur le problème d'utilité (Utility Deletion : UD)* [Minton, 1990], lorsque la BC atteint un seuil limite prédéfini, cette stratégie supprime les cas qui ont une valeur de *l'utilité négative*.
Le problème d'utilité est défini par la différence entre la *taille* de la BC, la *qualité* de la solution et le problème *d'efficacité* associée aux grandes BC. *L'efficacité* du système est mesurée par le temps moyen mis pour résoudre un problème cible. La réduction du temps de la solution correspond à l'augmentation de l'efficacité. *La qualité* de la solution est lié au pourcentage de bonnes réponses fournit par le système. Elle augmente avec la taille de la BC [Smyth et al., 1996].
- *Suppression basée sur la redondance et l'inconsistance* [Racine et al., 1996], [Racine et al., 1997], cette méthode est dotée de deux modules de détection l'un de *redondance* et l'autre de *l'inconsistance*. Suivant une série de test de chaque cas de la BC par ces deux modules, ceux-ci sont supprimés ou non de la BC avec l'approbation de l'utilisateur.
- *Suppression basée sur la taille et la densité de la BC*, cette stratégie de suppression a été proposée par B. Smyth et M.T. Keane [Smyth et al., 1998b] et étudie la taille de la BC, la densité et la distribution des cas dans la BC. Elle essaye de garder l'homogénéité de la densité des cas dans la BC.
La Densité de la BC est l'ensemble de densité des cas dans la BC, on dit qu'un groupe est dense lorsque la similarité entre les cas est grande dans la résolution de problème dans un groupe de cas défini.
La Distribution de la BC est la façon dont les cas sont organisés dans la BC. Une BC mal distribuée est une BC qui aura une variation considérable de densité, par conséquent, une irrégularité dans la solution donnée aux problèmes.

4.2. Méthodes à partir de catégorisation des cas

Ces méthodes font appel à une modélisation de la compétence de la BC qui ont été proposés par B. Smyth et M.T. Keane [Smyth et al., 1995], [Smyth et al., 1998b], [Smyth et al., 1999a], [Smyth et al., 2002a], [Smyth et al., 2002b]. Une catégorisation de cas dans la BC est créée selon leurs compétences. Cette catégorisation prend appui sur deux notions importantes qui sont *le recouvrement (coverage)* et *l'atteignabilité (reachability)*.

Soit une base de cas $C = \{c_1, \dots, c_n\}$ et $c \in C$ l'ensemble des cas cibles dans la BC.

L'ensemble de recouvrement d'un cas représente l'ensemble de cas cibles qu'il peut résoudre.

- Pour $c \in C$, $\text{Recouvrement}(c) = \{c' \in C : \text{résout}(c, c')\}$

L'ensemble d'atteignabilité d'un cas cible est l'ensemble de cas qui peuvent être utilisés pour le résoudre.

- Pour $c \in C$, $\text{Atteignabilité}(c) = \{c' \in C : \text{résout}(c', c)\}$

Ainsi, 4 classes de cas sont considérées :

- *Pivot* : $\text{Pivot}(c) \text{ ssi } \text{Atteignabilité}(c) - \{c\} = \emptyset$
- *Support* : $\text{Support}(c) \text{ ssi } \exists c' \in \text{Atteignabilité}(c) - \{c\} : \text{Recouvrement}(c') \subset \text{Couverture}(c)$
- *Couverture* : $\text{Couverture}(c) \text{ ssi } \text{Pivot}(c) \wedge \text{Recouvrement}(c) \cap \bigcup_{c' \in \text{Atteignabilité}(c) - \{c\}} \text{Recouvrement}(c') \neq \emptyset$
- *Auxiliaire* : $\text{Auxiliaire}(c) \text{ ssi } \exists c' \in \text{Atteignabilité}(c) - \{c\} : \text{Recouvrement}(c) \not\subset \text{Recouvrement}(c')$

Les méthodes développées dans ce cadre génère un ensemble de cas cibles afin de catégoriser la BC. Deux hypothèses de base sous tendent ces modèles à savoir la BC correspond à un échantillon de cas cibles ou cas potentiels et l'espace de problèmes est régulier, ce qui veut dire que les problèmes similaires ont des solutions similaires.

- *Suppression de traces (footprint deletion : FD)*, cette stratégie de suppression de cas vise à maximiser la compétence en minimisant la taille de le BC en la guidant vers une configuration optimale. Cette BC optimale est appelée *trace de compétence*. Cette stratégie s'appuie sur la catégorisation des cas de la BC pour pouvoir supprimer tous les cas dit auxiliaires, de couverture et de ne laisser qu'un cas de support dans chaque groupe de support et de garder tous les cas pivots. Cette stratégie préserve la compétence et ne s'occupe pas du problème de la performance.
- *Suppression trace-utilité (footprint-utility deletion : FUD)* [Smyth et al., 1995], Cette stratégie améliore la précédente en tenant compte de la performance des différents cas quand il faut en garder un parmi d'autres à éliminer. Elle s'occupe conjointement de la compétence et de la performance.

5. Proposition de méthodes de modélisation de la compétence de la Base de Cas

5.1 Proposition en utilisant des méthodes d'apprentissage automatique

Notre travail est destiné à maintenir une B.C de diagnostic industriel qui peut se formaliser sous forme d'exemples, ou la classe à retrouver est une classe d'équipement à réparer. On peut donc se permettre d'appliquer des algorithmes d'apprentissage supervisé.

On propose trois étapes comme dans le cas de Simon [Simon et al., 2001].

1° On applique sur la B.C un algorithme de sélection de variables qui permet de détecter des variables symboliques corrélées [Chebel-Morello et al., 2001].

2° Puis nous appliquons à cet ensemble de cas restreint par le nombre de variables, et d'objets, un arbre de décision C4.5 en l'occurrence

3° On met en place une stratégie de suppression de cas visant à maximiser la compétence en minimisant la taille de le BC, reposée sur la classification de quatre types de cas.

- a- Pour cela on sélectionne la profondeur d'arbre. Si on a des feuilles à cette profondeur, on ne conservera qu'un seul cas qui sera un cas pivot.
- b- Si on a des nœuds à cette profondeur. Les cas de ce nœud seront un groupe de cas de support. Les variables supplémentaires, en dessous de ce noeud permettront de construire les règles d'adaptation permettant de retrouver les cas se trouvant dans la population associée au nœud élagué, à partir du cas de support.

5.2. Proposition d'une stratégie de suppression de cas basée sur la compétence

Nous nous sommes intéressés au développement du critère de la compétence concernant les stratégies de suppression de cas car aucune métrique dans cette stratégie n'a été développée. Par contre trois métriques ont été développées dans la stratégie d'ajout de cas, celle du RC pour la compétence et celle du RP et du PB pour la performance.

Notre méthode est une combinaison de la méthode de Smyth et Mckenna (établie sur des notions du recouvrement et d'atteignabilité) et de la mesure du recouvrement relatif RC (qui est utilisée dans la stratégie d'ajout de cas [Smyth et al., 1999b]).

5.3. La mesure de la compétence (MC) pour la sélection des cas

Lorsqu'on parle de la compétence d'un cas, on se réfère à son pouvoir de résolution sur un certain nombre de problèmes cibles.

On considère une BC « C » contenant un certain nombre de cas « c_i » : $C = \{c_i \mid i \in (1, \dots, n)\}$

Lorsqu'on veut calculer la compétence d'un cas « c_k » dans la BC, on considère l'espace des cas cibles tous les cas appartenant à la BC y compris le cas « c_k » en question.

Notre mesure (MC) se base essentiellement sur les deux notions de *recouvrement* et d'*atteignabilité*.

Soit un exemple d'une BC contenant quatre cas, montrant l'espace de recouvrement et d'atteignabilité des cas c₁, c₂, c₃ et c₄ (voir *figure 2* reprise des travaux de [Smyth et al., 1998b]).

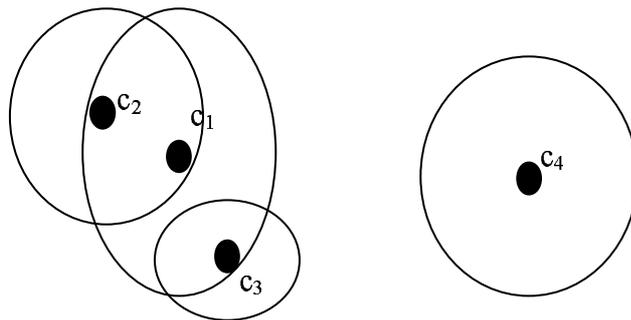


Figure2. Une BC contenant quatre cas {c₁, c₂, c₃, c₄} avec leur espace de recouvrement et d'atteignabilité.

On attribut à chaque recouvrement et atteignabilité d'un cas une valeur qu'on notera *valeur recouvrement* « Vr » et *valeur atteignabilité* « Va ».

Recouvrement(c ₁) = {c ₁ , c ₂ , c ₃ } → Vr(c ₁)=3	Atteignabilité(c ₁) = {c ₁ , c ₂ } → Va(c ₁)=2
Recouvrement(c ₂) = {c ₁ , c ₂ } → Vr(c ₂)=2	Atteignabilité(c ₂) = {c ₁ , c ₂ } → Va(c ₂)=2
Recouvrement(c ₃) = {c ₃ } → Vr(c ₃)=1	Atteignabilité(c ₃) = {c ₁ , c ₃ } → Va(c ₃)=2
Recouvrement(c ₄) = {c ₄ } → Vr(c ₄)=1	Atteignabilité(c ₄) = {c ₄ } → Va(c ₄)=1

On définit une mesure (MC) devant estimer la contribution individuelle de la compétence d'un cas « c_k » en fonction de la taille de l'ensemble du recouvrement de ce dernier.

$$\text{« MesureCompétence}(c) = \frac{Vr(c)}{Va(c)} \text{ »}$$

Cette mesure est utilisée pour guider la suppression des cas dans la BC « Favoriser les cas qui possède une grande valeur de 'MC' et supprimer ceux qui ont une petite valeur »

5.4. Méthode de sélection et de suppression des cas

Cette méthode suit le principe des stratégies de suppression de cas, c'est à dire *en prenant une BC complète telle qu'elle est, ensuite on procède à la suppression selon la valeur de « MC » conformément aux étapes suivantes :*

Soit une BC contenant des cas ainsi que l'ensemble de cas cibles définit précédemment.

- a. Calcul de Vr(c_i) et Va(c_i) et créer des groupes de recouvrement.
- b. Calcul de MC(c_i)
- c. Suppression des cas suivant leurs valeurs de « MC »

Le détail de l'étape « a » est décrit ci-dessous :

Algorithme de calcul de la valeur du recouvrement des cas de la BC

Pour tout $cas_i \in BC$

Pour tout $cas_j \in BC$

Si cas_i apporte une solution à cas_j

 Augmenter valeur Vr_i

 Mémoriser l'index du cas cas_j

Fin si

Fin Pour

$cas_i \leftarrow (Vr_i, \text{index } cas_j, \text{classe})$

Fin Pour

Par analogie, on applique ce même principe pour calculer la valeur de $Va(c_i)$.

De ce fait, lorsqu'on applique les deux algorithmes sur l'ensemble des cas dans la BC, on obtiendra pour chaque cas sa valeur $Vr(c)$, $Va(c)$ ainsi que les différents groupes de recouvrement. A partir de ces deux valeurs, on peut caractériser les cas et garder les cas pivots.

Propriétés

- *Quand $Va(c_i) = 1$ et $MC(c_i) = 1 \rightarrow$ cas pivot (selon Smyth et Mckenna)*
- *Quand $Va(c_i) > 1$ et $MC(c_i) = 1 \rightarrow$ suppression du cas (cas auxiliaire)*
- *Quand on a un groupe de cas où il y'a que des cas qui ont les mêmes $Vr(c_i)$ et $Va(c_i) > 1$, donc les même $MC \rightarrow$ suppression de tous les cas de ce groupe en ne laissant qu'un seul cas (cas de supports).*

Exemple

Soit une BC contenant quatre cas $C = \{c_1, c_2, c_3, c_4\}$ (figure2). On calcule pour chaque cas MC selon leurs valeurs de Vr et Va .

$MC(c_1) = 1.5$, $MC(c_2) = 1$, $MC(c_3) = 0.5$, $MC(c_4) = 1$.

- On constate que $MC(c_2) = MC(c_4) = 1$. Ce sont deux cas complètement différents. Le cas c_4 est un cas pivot, on supprime donc le cas c_2 .

- On constate aussi que $MC(c_1) > MC(c_3) \rightarrow$ suppression du cas c_3 .

On obtient donc la suppression de deux cas (c_2 et c_3) et une BC contenant les cas (c_1 et c_4).

En recalculant la valeur de MC pour chaque cas on trouve $MC(c_1) = MC(c_4) = 1$ avec $Va(c_1) = Va(c_4) = 1$ et $Vr(c_1) = Vr(c_4) = 1$.

Ainsi on obtiendra une BC réduite possédant le moins de cas possible et un recouvrement maximal.

La valeur de la compétence globale de la BC réduite est égale à la somme des MC des cas dans cette BC divisée par le nombre de cas présents.

$$\ll \text{CompétenceGlobale}(BC) = \frac{\sum_{i=1}^n MC(c_i)}{\sum_{i=1}^n c_i} \gg$$

La BC réduite est celle qui a pour valeur de la $\text{CompétenceGlobale}(BC) = 1$. (C'est le critère d'arrêt).

L'objectif visé par notre mesure est de calculer le taux de recouvrement et d'atteignabilité des cas suivant un algorithme simple de classification de cas en les indexant suivant leurs différentes valeurs calculées de Va et Vr .

La mesure cherche à maximiser la compétence de la BC donc le recouvrement et à minimiser l'atteignabilité des cas dans la BC. Le critère d'arrêt de notre méthode, contrairement à la méthode RC-CNN tient compte de la compétence globale de la BC et non pas d'un seuil fixé arbitrairement (comme le nombre de cas de la BC).

6. Conclusions

Actuellement il existe un certain nombre de travaux sur les systèmes à base de connaissances en général les systèmes experts et en particulier les systèmes de raisonnement à partir de cas. Différentes approches ont été développées et concernent aussi bien l'insertion des opérations de maintenance dans le cycle de vie du système de conception des SBC, que des méthodes vérifiant la qualité de la connaissance utilisée. Ces méthodes peuvent être empruntées aux travaux d'apprentissage, ou aux méthodes de recherche sur la validation et la vérification des systèmes ou être spécifique aux travaux de RàPC. Dans les recherches faites dans la communauté de RàPC, une majorité des travaux est consacrée à la MBC. Ces travaux ont comme objectif d'assurer la compétence, la performance et le maintien dans le temps de la qualité des systèmes de RàPC. Parmi les politiques et stratégies de MBC existantes, nous proposons deux pistes de travail exploitant la stratégie de suppression de cas proposée par Smith. Nous prenons appui sur la caractérisation faites des différents cas dans la base, et nous retrouvons celle-ci soit par l'exploitation d'algorithmes d'apprentissage supervisé, soit par une autre méthode prenant en compte le critère de compétence de la BC. De cette classification, la suppression de cas est simple à réaliser. La notion de compétence sera quantifiée par une mesure de maintenance MC , qui pour être calculé tient compte de la méthode de Smyth et Mckenna établie sur des notions de recouvrement et d'atteignabilité et d'une mesure du recouvrement relatif RC. Cette mesure a l'avantage de tenir compte de la compétence et de la performance. Mais ce travail n'est qu'à ses débuts et demande à être poursuivi.

Références bibliographiques

- A. Aamodt et E. Plaza, (1994). Case-based reasoning: Foundational issues, methodological variation, and systems. *AI Communication*, 7(1) : 36-59.
- S.S. Anand, D. Patterson, J.G. Hughes et D.A. Bell, (1998). Discovering case knowledge using data mining. In *Proceedings of the 2nd Pacific Asia Conference on Knowledge Discovery and Data Mining: Current Issues and New Applications*, PAKDD-98. Springer-Verlag, Berlin, pp. 25-35.
- J.R.K.De. Basak et S.K. Pal, (1998). Unsupervised feature selection using a neuro-fuzzy approach. *Pattern Recognition Letters*, 19:998-1006.
- B. Bartsch-Spörl, D. Janetzko et S. Wess, (1995). Case-Based Reasoning for Decision Support and Diagnostic and Problem Solving: The INRECA Approach. *Proc. Of the 3rd German Workshop on CBR*, March 1995, LSA-Report 95-02, Centre for Learning System and Applications, University of Kaiserslautern, pp. 63-72.
- B. Chebel-Morello, D. Michaut, et P.Baptiste, (2001). A knowledge discovery process for a flexible manufacturing system. *Proc of the Emerging Technologies and Factory Automation*, october, 2001 Antibes, (1) pp 652 – 659.
- F. Coenen, B. Eaglestone et M. Ridley, (1999). Verification, Validation and Integrity in Knowledge and Database Systems: Future Directions. *EUROVAV 1999*: 297-312.
- T. Gabel, (2004). On the Use of Vocabulary Knowledge for Learning Similarity Measures.
- K. Hanney et M.T. Keane, (1996). Learning adaptation rules from a case-base. *Proceedings of the 3rd European Workshop on CBR* (pp. 179-192). Lausanne: Springer.

- I. Iglezakis et T. Roth-Berghofer, (2000). A survey regarding the central role of the case base for maintenance in case-based reasoning.
- D.B. Leake et D.C. Wilson, (1998). Categorizing case-base maintenance: dimensions and directions. *Advances in Case-Based Reasoning. 4th European Workshop, EWCBR 98*. Proceedings. 1998: 196-207, 1998. Springer-Verlag, Berlin, Germany.
- D.B. Leake et D.C. Wilson, (2000a). Remembering Why To Remember: Performance-guided case-base maintenance. *Advances in Case-Based Reasoning: Proceeding of EWCBR-2K*, 2000. Springer-Verlag.
- D.B. Leake et D.C. Wilson, (2000b). Guiding Case-Base Maintenance: Competence and Performance. *Online Proceedings of the ECAI'2000 Workshop on Flexible Strategies for Maintaining Knowledge Containers*.
- M. Malek, (2000). Hybrid approaches for integrating neural networks and case based reasoning : From loosely coupled to tightly coupled models. In *Soft Computing in Case Based Reasoning*, éditeurs Tharam S. Dillon Sankar K. Pal et Daniel S. Yeung, pages 73-94.
- S. Markovitch et P.D. Scott, (1988). The Role of Forgetting in Learning. In Proceedings of the Fifth International Conference on Machine Learning, pages 459-465.
- S. Minton, (1990). Qualitative Results Concerning the Utility of Explanation-Based Learning. *Artificial Intelligence*, 42: 363-391.
- M. Nick et K-D. Althoff, (2000). Systematic Evaluation and Maintenance of Experience Bases. *Workshop on Flexible Strategies for Maintaining Knowledge Containers (WS30 at ECAI)*.
- K. Racine et Q. Yang, (1996). On the consistency Management of Large Case Bases: the Case for Validation. *To appear in AAAI Technical Report. Verification and Validation Workshop*.
- K. Racine et Q. Yang, (1997). Maintaining Unstructured Case Bases. Case-based reasoning research and development. *Lecture notes in computer science: 553-564*. Providence RI, July 25-27.
- M.M. Richter, (1995). The knowledge contained in similarity measures. Invited talk at ICCBR-95. Valable sur <http://www.arg.informatil.uni-kl.de/~Isa/CBR/Richtericcbr95remarks.html>.
- M.M. Richter, (1998). Introduction. In Case-Based Reasoning Technology: From Foundations to Applications. Edited by M. Lena, B. Bartsc-Sporl, H. D. Burkhard, et S. Wess. Springer-Verlag, Berlin, pp. 1-15.
- T.H. Roh, K.J. Oh, et I. Han, (2003). The collaborative filtering recommendation based on som clustering-indexing cbr. In *Expert Systems with Applications*, volume 25, pages 413-423. Elsevier Science.
- T. Roth-Berghofer et I. Iglezzakis, (2001). Six Steps in Case-Based Reasoning: Towards a maintenance methodology for case-based reasoning systems. *Includes Proceedings of the 9th German Workshop on CBR, GWCBR*, Germany,
- C.K.S. Simon et D.S Yeung, (2001). Transferring case knowledge to adaptation knowledge: an approach for case-base maintenance. *Computational Intelligent*, Volume 17, Number 2.
- B. Smyth et M.T. Keane, (1995). Remembering To Forget: A competence Preserving Deletion Policy for Case-Based Reasoning Systems. In : *Proceeding of the 14th International Joint Conference on Artificial Intelligent*. Morgan-Kaufmann. 377-382.
- B. Smyth et Pádraig Cunningham, (1996). The Utility Problem Analysed: A Case-Based Reasoning Perspective. *Third European Workshop on Case-Based Reasoning*. Lausanne, Switzerland.
- B. Smyth, (1998a). Case-base maintenance. Tasks and Methods in Applied Artificial Intelligence. *11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA-98-AIE*. Proceedings. 1998: 507-516 vol.2, 1998. Springer-Verlag, Berlin, Germany.
- B. Smyth et E. McKenna, (1998b). Modelling the Competence of Case-Bases. *Advances in case-based reasoning. Lecture notes in computer science*, 1488: 208-220. Dublin, 23-25.
- B. Smyth et E. McKenna, (1999a). Footprint-Based Retrieval. In *Proceedings of the Third International Conference on Case-Based Reasoning, ICCBR '99*. Berlin, 27-30 July 1999. Springer Verlag, Germany.
- B. Smyth et E. McKenna, (1999b). Building Compact Competent Case-Bases. Case-based reasoning research and development. *Lecture notes in computer science*. 1650: 329-342. See on Monastery, 27-30 July 1999.
- B. Smyth et E. McKenna, (2002a). Competence guided incremental foot-print. *Journal of Knowledge-Based Systems*, In press.

- B. Smyth et E. McKenna, (2002b). Competence models and the maintenance problem. *Computational Intelligence: Special Issue on Maintaining Case-Based Reasoning Systems*, In Press.
- D. Wettscherk et D.W. Aha, (1995). Weighting features. In *proceedings of the 1st International Conference of Case-Base Reasoning, ICCBR-95*. Springer-Verlag, Berlin, pp.347-358.
- Q. Yang et J. Wu, (2000). Keep it simple: A case-base maintenance policy based on clustering and information theory. *13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2000: advances in artificial intelligence. Proceedings Lecture Notes in Artificial Intelligence*, Vol.1822. 2000: 102-114. Montréal PQ, 14-17 May 2000. Springer-Verlag, Berlin, Germany.
- Q. Yang et J. Zhu, (2001). A case addition policy for case-base maintenance. *Computational Intelligence Journal, A Special Issue on Case-Base Maintenance*. Blackwell Publishers, Boston MA UK, 17(2) : 250-262.
- F. Zehraoui (2004). Systèmes d'apprentissage connexionnistes et raisonnement à partir de cas pour la classification et le classement de séquences. *Thèse de doctorat*, université Paris-Nord. 12 Octobre 2004.