Maintaining Footprint-Based Retrieval for Case Deletion

Ning Lu, Jie Lu, Guangquan Zhang

Decision Systems & e-Service Intelligence (DeSI) Lab
Centre for Quantum Computation & Intelligent Systems (QCIS)
Faculty of Engineering and Information Technology, University of Technology,
Sydney P.O. Box 123, Broadway, NSW 2007, Australia
Email: {philiplu, jielu, zhangg}@it.uts.edu.au

Abstract. The effectiveness and efficiency of case-based reasoning (CBR) systems depend largely on the success of case-based retrieval. The case-base maintenance (CBM) issues become imperative and important especially for modern societies. This paper proposes a new competence model and a new maintenance procedure for the proposed competence model. Based on the proposed competence maintenance procedure, footprint-based retrieval (FBR), a competence-based case base retrieval method, is able to preserve its own retrieval effectiveness and efficiency.

Keywords: case-based reasoning, case-based maintenance, footprint-based retrieval, competence models

1 Introduction

Case-based reasoning (CBR) systems solve new problems by retrieving and adapting solutions of previously solved problems that have stored in a case base. Obviously, the success of case-based problem solving is critically dependent on the retrieval of suitable cases [1]. Generally, a retrieval method is the combination of two procedures: a similarity assessment procedure to determine the similarity between a given case and target problem, and a procedure for searching the case memory in order to find the most similar cases [2]. Correspondingly, research on case base retrieval has two folds, with one fold focus on developing more efficient and accurate similarity assessment [1, 3, 4] and another focus on reducing the search needed to locate the best case without degrading competence or quality, such as footprint-based retrieval (FBR) [2], *k*d-tree [5], condensed nearest neighbor (CNN) [6]. Although these searching methodologies return suitable results and dramatically decrease the average searching time required, they tend to rely on some kinds of pre-structuring of the case memory.

Recently, the case base maintenance (CBM) issue has drawn more and more attention to the CBR community. The term CBM refers to the process of refining a CBR system to improve the system's performance, which implements policies of revising the contents of a CBR system in order to facilitate future reasoning [7]. For example, case deletion policy [8] or case addition policy [9, 10] is conducted to manage the case base knowledge in a CBR system. As a result of such changes in case memory, we also noticed a requirement for updating the indexing structures adopted in case base retrieval. Improperly managing of the indexing structure may degrade the retrieval efficiency or even fail to solve the target problem. For example, frequently changes in a case base may finally result an unbalanced tree, the deletion of cases in the reference set [6] may lead to unsuitable cases retrieved. Although the problem could be simply solved by restructuring with the whole case memory each time changes are made to the case base, it posts an unnecessary and undesirable cost to the CBR cycle, especially when changes are frequent and the restructuring cost is high. Thus it is imperative to develop new maintaining methods for case base retrieval.

In this paper, we suggest a new competence updating procedure for maintaining FBR [2]. Our work distinguishes from a previous work [11] in two aspects. First, we replace the *competence group* [12] with a new model - *competence closure*. We compare the differences between these two models, and show that our new competence model can still applied to the FBR. Second, we develop a new procedure for updating the competence model for case deletion which could be seen as a counterpart of Smyth and McKenna's work [11] which devoted to the scenario of case addition. Our work will contribute the research in the area of competence-based CBM.

This paper is organized as follow. Section 2 reviews the related works concerning the case base competence model [12] and FBR [2]. In Section 3, we first propose our new competence model, and then give out our algorithms of the proposed updating procedure. An example is given in Section 5 to illustrate the proposed updating procedure. Conclusions and future works come in Section 6.

2 Related Works

2.1 A Competence Model

According to Smyth and McKenna [12], the local competence of individual cases is characterized by its *coverage* and *reachability*. The *coverage* of a case is the set of target problems that it can be used to solve. The *reachability* of a target problem is the set of cases that can be used to provide a solution for the target. Since it is impossible to enumerate all possible future target problems, in practice the *coverage set* of a case is estimated by the set of cases that can be solved by its retrieval and adaption. And the *reachability set* of a case is estimated by the set of cases that can bring about its solution (see Definitions 1 and 2 respectively).

Definition 1: [12]

$$\text{Given a case base } \mathrm{C} = \{\mathrm{c}_1, \dots, \mathrm{c}_\mathrm{n}\}, \text{For } \mathrm{c} \in \mathrm{C},$$
$$\mathrm{CoverageSet(c)} = \{\mathrm{c}' \in \mathrm{C}: \mathrm{Solves(c, c')}\}$$

Definition 2: [12]

$$\mathrm{ReachibilitySet(c)} = \{\mathrm{c}' \in \mathrm{C}: \mathrm{Solves(c', c)}\}$$

The *coverage set* and *reachability set* only provide a local estimation of competence. In order to estimate the true competence contributions of all cases in case base as a whole, the interactions between cases in terms of how their coverage and reachability sets overlap can further be modelled by *related set*, *shared coverage* and *competence group* [12]. The *related set* of a case is defined as the union of its *coverage set* and *reachability set* (Def. 3). The *shared coverage* of two cases exists if and only if the intersection of the *related sets* of two different cases is not empty (Def. 4). Finally all cases in a case base can be uniquely grouped together into *competence groups* (Def. 5) which are maximal sets of cases exhibiting shared coverage.

Definition 3: [12]

$$\mathrm{RelatedSet(c)} = \mathrm{CoverageSet(c)} \cup \mathrm{ReachabilitySet(c)}$$

Definition 4: [12]

$$\begin{array}{l}\text{For } \mathrm{c_1, c_2} \in \mathrm{C, SharedCoverage(c_1, c_2)} \\ \qquad \text{iff } [\mathrm{RelatedSet(c_1)} \cap \mathrm{RelatedSet(c_2)}] \neq \emptyset\end{array}$$

Definition 5: [12]

$$\begin{array}{l}\text{For } \mathrm{G} = \{\mathrm{c_1} \cdots \mathrm{c_n}\} \subseteq \mathrm{C}, \qquad \mathrm{CompentenceGroup(G)} \\ \qquad \text{iff } \forall \mathrm{c_i} \in \mathrm{G}, \exists \mathrm{c_j} \in \mathrm{G} - \{\mathrm{c_i}\}: \mathrm{SharedCoverage}(\mathrm{c_i}, \mathrm{c_j}) \wedge \\ \qquad\qquad \forall \mathrm{c_k} \in \mathrm{C} - \mathrm{G}, \nexists \mathrm{c_l} \in \mathrm{G}: \mathrm{SharedCoverage(c_k, c_l)}\end{array}$$

2.2 Footprint-based Retrieval

Based on the competence model (Section 2.1), the *footprint set* of a case base is further defined as a minimal subset of the case base that covers all the cases in the case base with regarding to its competence [2]. It is constructed by computing *group footprint* for each *competence group* of a case-base.

Footprint-based retrieval (FBR) [2] is a case base retrieval method by using footprint cases and their *related sets*. The key idea of FBR is to dynamically find a subset of the original case base which closely related to the target problem, and search the best case within it. There are two separated stages required for FBR, with stage one, searching a reference case from the footprint set in order to identify a local region of the case-base that similar to the target problem and stage two, retrieving from the related set of the reference case to locate the best case for the target problem. Although FBR dramatically decreases its average retrieval time, while preserving high retrieval optimality, its success largely depends on a well established footprint set and a sound competence model of the case base. However, without proper methods to manage and maintain the footprint set and the competence model, the effectiveness and efficiency of FBR deteriorates.

3 A Complementary Procedure for Updating Competence Model

Similar to case addition [11], the deletion of a case in the case base should also trigger the update of the competence model. Thus to develop a procedure for updating competence model when deleting a case in the case base, we suggested a new competence model – *competence closure*. Based on this model, we developed an updating method for case deletion which contains three steps: update local competences, replace competence closure and setup footprint set.

3.1 A New Competence Model

Since a *competence group* (Def. 5) must contain at least two cases, this model may not be able to represent all cases inside a case base, when the case base contains some rare cases, or a case is removed from an existing *competence group* with only two cases. In addition, a case base maybe not able to be exclusively grouped by all competence groups it contained, which may cause problems when implementing this model in real world. In this sense, we suggest a new competence model - *competence closure* which is defined as the maximal set of cases linked together though their *related set* (Def. 6). We show the differences between these two models in Figure 1.

Definition 6:

$$\text{For } \mathrm{G} = \{\mathrm{c}_1 \cdots \mathrm{c}_\mathrm{n}\} \subseteq \mathrm{C},\ \text{CompentenceClosure (G)}$$
$$\text{iff } \forall \mathrm{c}_\mathrm{i}, \mathrm{c}_\mathrm{j} \in \mathrm{G}, \text{if } \mathrm{c}_\mathrm{i} \neq \mathrm{c}_\mathrm{j}, \exists \left\{\mathrm{c}_{\mathrm{i}_1}, \mathrm{c}_{\mathrm{i}_2}, \cdots, \mathrm{c}_{\mathrm{i}_\mathrm{k}}\right\} \subseteq \mathrm{G}\,,$$
$$\text{st. SharedCoverage}\left(\mathrm{c}_{\mathrm{i}_\mathrm{j}}, \mathrm{c}_{\mathrm{i}_{\mathrm{j}+1}}\right) \neq \emptyset\ (\mathrm{j} = 0, \cdots, \mathrm{k}) \text{ where } \mathrm{c}_\mathrm{i} = \mathrm{c}_{\mathrm{i}_0}, \mathrm{c}_\mathrm{j} = \mathrm{c}_{\mathrm{i}_{\mathrm{k}+1}}\ \wedge$$
$$\forall \mathrm{c}_\mathrm{k} \in \mathrm{C} - \mathrm{G}, \nexists \mathrm{c}_\mathrm{l} \in \mathrm{G} \quad \text{st. SharedCoverage}(\mathrm{c}_\mathrm{k}, \mathrm{c}_\mathrm{l}) \neq \emptyset$$

Based on our new competence model, the black cases in a case base C can be linked together with regarding to their shared coverage. Finally, all cases will be uniquely grouped into several *competence closures*. A *competence group* can be formed by the union of any *competence closures* with at least two cases. In this sense, a *competence closure* is also a *competence group*, if and only if it has at least two cases. But a *competence group* is not necessarily to be a *competence closure.* In addition, a rare case can setup its own *competence closure* (CC_3) if it has no similar cases, but this case cannot be modeled by the *competence group*. Clearly, our new competence model can be used to replace the *competence group* in FBR. By doing that, we are able to model all cases in a case base.

3.2 An Updating Algorithm

The competence model of a case base will be affected differently when different cases are removed. Based on whether the deleted case is a footprint case or not and whether the deletion of a case will cause splitting of existing *competence closure*, there are four possible situations when deleting a case in case base. When deleting a non-footprint case and no change to the *competence closure* is required, we only update the local competence estimation of those cases in the related set of the removed case. Actually, this could be seen as the basic scenario, since the local competence will be updated for all the other situations as well. When deleting a non-footprint case and new *competence closures* are created, we reserve the current *footprint set*, while assigning each footprint case of the original *competence closure*, which the removed case belongs to, to the right *competence closure* which it belongs to. When deleting a footprint case and no change to the *competence closure* is required, we replace the *footprint set* of current *competence closure,* which the removed case belongs to. When deleting a footprint case and new *competence closures* are created, we reconstruct both the *competence closures* and their corresponding *footprint sets*.

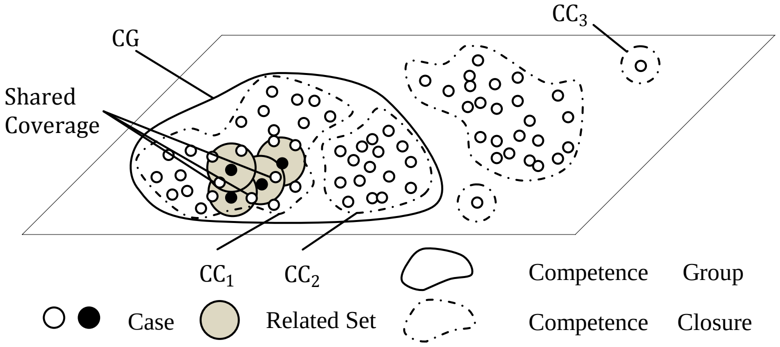

Fig. 1. Procedure for Update Local Competences

3.2.1 Update Local Competences

Different from the scenario of case addition, updating the local competence models of the remaining cases in the case base does not necessarily require checking the whole case base, but only the related ones. According to Def. 1 and Def. 2, we noted that for any case - *c*, a case - *x* belongs to the *coverage set* of *c*, if and only if *c* belongs to the *reachability set* of *x* (Theorem 1). In this sense, only the local competence of the cases in the *related set* of the deleted case will be affected. Thus we give out the update procedure of local competence for deletion (Algorithm. 1).

Theorem 1:

$$\text{Given } \forall c \in C = \{c_1, c_2, \dots, c_n\},\ \text{ then we have } x \in \text{CoverageSet}(c)$$
$$\text{iff } c\ \in \text{ReachibiltySet}(x)$$

```
c ← deleted case
For each case x ∈ CoverageSet(c)
      Remove c from ReachabilitySet(x)
EndFor
For each case x ∈ ReachabilitySet(c)
      Remove c from CoverageSet(x)
EndFor
```

Algorithm 1. Updating Local Competences for Case Deletion

3.2.2 Replace Competence Closure

As the competence closure also changes along with the deletion of cases. Current competence closure containing the deleted case should be updated or replaced when new competence closures are created. We check whether the current competence closure still holds while setting up new groups when it does not (Algorithm. 2).

```
G ← competence closure, that c ∈ G
Remove c from G
replace-competence-Closure(G) {
      Create a case set m
      Get a case x from G
      Create a new competence closure G′ = G - {x}
      Add x into m
      while (m is not ∅)
            Get a case y from m
            For each case z ∈ RelatedSet(y)
                  If z ∈ G′
                        Add z into m
                        Remove z from G′
                  EndIf
            EndFor
            Remove y from m
      EndWhile
      If G′ is not {}
            Add G-G′ as a new competence closure
            Set G = G′
            call replace_competence_Closure(G)
      EndIf
}
```

Algorithm 2. Replacing Competence Closure for Case Deletion

3.3.3 Setup Footprint Set

Since the deletion of a non-footprint case will only affect the competence model behind the footprint set, but does not degrade the coverage of current footprint set. We assign each *competence closure* with its corresponding footprint set, but do not actually change the footprint set with regarding to the whole case base. However, if a case was deleted from the footprint set, we setup new footprint set for each *competence closure* built in Step 2 (Section 3.2.2). The algorithm of constructing footprint sets is given in Algorithm 3.

4 A Case Study

The deletion of cases from the case base will trigger the process of updating the competence model, which in turn may cause the replacement of current competence closure and selected footprint cases. In this section, we illustrate a simple scenario of deleting a non-footprint case, which finally results to the splitting of a competence closure, but remaining the footprint set unchanged.

As shown in Figure 2, cases in the case base have been grouped into two *competence closures*, and footprint cases have been chosen to facilitate case-base retrieval. When a non-footprint case is removed from the case base, the local competence model will be first updated. In our example, removing the case will lead to cases in a *competence closure* no longer linked all together through their shared coverage. Thus new *competence closures* will be created to replace the original one. Correspondingly, the selected footprint cases will be assigned to the new *competence closures* which they actually belong to.

```
CC ← competence closures used to replace G
FP ← FootprintSet(G)
If c ∈ FP
      For each competence closure g ∈ CC
            Setup footprint set for g
      EndFor
Else
      For each case x ∈ FP
            Find competence closure g∈CC, that x∈g
            Remove x from FootprintSet(G)
            Add x to FootprintSet(g)
      EndFor
EndIf
```

Algorithm 3. Setting up Footprint Set for Case Deletion

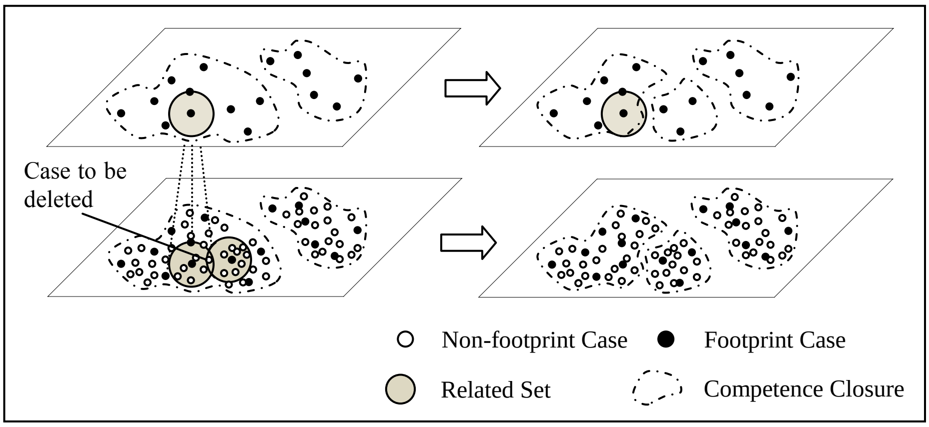

Fig. 2. Example of Deleting a Non-footprint Case

5 Conclusions and Future works

Modern case base maintenance issues post challenges to the traditional case base retrieval methods. A certain case base retrieval method degrades its performance without proper maintenance. In this paper, we have proposed a new competence model - *competence closure*, and a new maintenance procedure for case deletion based on this new model. With our new model and the proposed procedure, FBR is able to maintain the competence model on which it based by itself, thus preserve its retrieval effectiveness and efficiency.

For future work, characteristics and relationships between our new competence model and Smyth and McKenna's models [12] still need further investigation. Also how to maintain a case base retrieval method when other forms of knowledge in a case base change will be another task.

Acknowledgment. The work presented in this paper was supported by Australian Research Council (ARC) under Discovery Project DP0880739.

References

1. Smyth, B. and M.T. Keane, Retrieving Adaptable Cases: The Role of Adaptation Knowledge in Case Retrieval, Selected papers from the First European Workshop on Topics in Case-Based Reasoning. 1994, Springer-Verlag.
2. Smyth, B. and M. Elizabeth, Footprint-Based Retrieval, Proceedings of the Third International Conference on Case-Based Reasoning and Development. 1999, Springer-Verlag.
3. Castro, J.L., et al., Loss and gain functions for CBR retrieval. Information Sciences, 2009. 179(11): p. 1738-1750.
4. Cheng, W. and E. Hüllermeier, Learning Similarity Functions from Qualitative Feedback, Advances in Case-Based Reasoning. 2008. p. 120-134.
5. Wess, S., K.-D. Althoff, and G. Derwand, Using k-d trees to improve the retrieval step in case-based reasoning, Topics in Case-Based Reasoning. 1994. p. 167-181.
6. Hart, P., The condensed nearest neighbor rule (Corresp.). Information Theory, IEEE Transactions on, 1968. 14(3): p. 515-516.
7. Wilson, D.C. and D.B. Leake, Maintaining Case-Based Reasoners: Dimensions and Directions. Computational Intelligence, 2001. 17(2): p. 196.
8. Smyth, B. and M.T. Keane. Remembering To Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence. 1995. San Francisco: Morgan Kaufmann.
9. Yang, Q. and J. Zhu, A Case-Addition Policy for Case-Base Maintenance. Computational Intelligence, 2001. 17(2): p. 250.
10. Zhu, J. and Q. Yang. Remembering to Add: Competence-preserving Case-Addition Policies for Case-Base Maintenance. Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI). 1999: Morgan Kaufmann.
11. Smyth, B. and E. McKenna, An Efficient and Effective Procedure for Updating a Competence Model for Case-Based Reasoners, Machine Learning: ECML 2000. 2000. p. 357-368.
12. Smyth, B. and E. McKenna, Competence Models and the Maintenance Problem. Computational Intelligence, 2001. 17(2): p. 235.