

An Efficient and Effective Procedure for Updating a Competence Model for Case-Based Reasoners

Barry Smyth and Elizabeth McKenna

Department of Computer Science, University College Dublin,
Belfield, Dublin 4, Ireland
{Barry.Smyth,Elizabeth.McKenna}@ucd.ie

Abstract. Case-based reasoning systems solve new problems by reusing previous problem solving experience stored as cases in a case-base. In recent years the maintenance problem has become an increasingly important research issue for the case-based reasoning community. In short, the goal is to develop strategies for effectively maintaining the efficiency and competence of case-based reasoning systems as they evolve. Our research has focused on the development of a model of competence for case-based reasoning systems, a model that measures the contributions of individual cases to overall system competence, and which forms the computational basis for a variety of maintenance strategies. However, while this model offers many potential advantages its upkeep adds an additional cost to the CBR cycle. In this paper we evaluate a new method for more efficiently updating the model at run-time.

1 Introduction

Case-based reasoning (CBR) systems solve new problems by retrieving and adapting the solutions of similar problems stored as cases in a case-base [3,4,10]. The CBR method has been successfully adapted for a wide variety of tasks (classification, diagnosis, prediction, planning, and design) across a wide variety of domains (for example, fraud detection, property valuation, route planning, and software design).

An important issue facing the CBR community involves the ongoing maintenance of CBR systems [5,7,8,11,12,13,14,15]. The maintenance problem focuses on the issue of how best to manage the organisation and contents of a case-base in order to optimise future reasoning performance with respect to an agreed set of performance objectives. In practice, this involves the development of a range of policies for controlling the growth of case-bases and the organisation and indexing of the cases themselves. For example, a variety of case addition and deletion policies have been proposed to manage case-base growth [9,11,13,15]. Ultimately these procedures encode judgements about the importance of cases with respect to the efficiency and competence of a given case-based reasoner, and then use these judgements to prioritise cases for addition or deletion.

Zhu and Yang [15] describe a procedure for case addition that uses a probabilistic estimate of competence (see [13] for a related approach). Smyth & Keane [11] have looked at techniques for deleting cases (mainly as a way of coping with the

deleterious performance effects of the so-called *utility problem* [2,6,9]). They pre-categorise the competence contributions of individual cases in order to prioritise cases for deletion with respect to competence, thereby guaranteeing the preservation of system competence during deletion. In contrast, coming from a speed-up learning background, Minton [6] proposes a technique for estimating the efficiency of knowledge items (cases) as a guide for an efficiency-based deletion criterion. Again, this is a coping strategy for dealing with the utility problem, but it is found to be inappropriate for CBR because it does not respect competence during deletion.

These approaches all focus on one particular aspect of the maintenance problem, for example case addition or deletion, and the result is a collection of solutions that are limited to particular instances of the more general maintenance problem. Furthermore, these solutions operate on the basis of key assumptions and heuristics about the nature of competence and efficiency in case-based reasoners, but they fail to reveal the full picture. As the core motivation for our research, we argue that any effective and generic solution to the maintenance problem will only come about by developing a more complete understanding of the true nature of efficiency and competence for case-based reasoning systems. In short, we propose the development of an explicit theory or model of competence to act as the foundation for maintenance solutions. We argue that such models will provide us with access to more accurate and effective measures of the “worth” or competence of a case in order to better inform future case deletion or addition strategies.

One such competence model is presented and evaluated in [12] where we show that it provides accurate predictions of true competence under a variety of operational conditions. In recent work we have also shown how this competence model can be used as the computational basis for a host of innovative approaches to case addition and deletion, case retrieval, authoring support and case-base visualisation [12,13,14]. However, while this model has proved to be effective at estimating case competence, as the case-base grows its update adds an additional cost to the CBR cycle. In this paper we explain how to significantly reduce this cost by presenting a novel procedure for more efficiently updating the competence model at run-time as new cases are learned (Section 3). In Section 4 we fully evaluate this new update procedure and in Section 5 we explain how further cost savings are possible by integrating the update procedure with an existing competence-guided retrieval technique; in fact, we argue that under certain conditions the competence update essentially comes for free as a side-effect of retrieval. We begin with an outline description of our existing competence model.

2 A Model of Case Competence

Our competence model can be best understood in terms of four distinct stages that combine local competence estimates to produce a global prediction of case-base competence. The following sections outline each of these stages and the interested reader is referred to [8,11,12,13,14] for further details.

2.1 Local Competence Estimates

The local competence contributions of individual cases are characterised by two sets. The *coverage set* of a case is the set of *target problems* that this case can successfully solve, while the *reachability set* of a *target problem* is the set of *cases* that can solve this target problem. It is impossible to enumerate all possible future target problems (T), but by using the case-base (C) itself as a representative sample of the target problem space we can efficiently estimate these sets as shown in Def. 1 and 2.

Def. 1 CoverageSet(c)={c'∈C: Solves(c,c')}

Def. 2 ReachabilitySet(c)={c'∈C: Solves(c',c)}

2.2 Shared Coverage & Competence Groups

Coverage and reachability sets are local estimates of competence only and to estimate the true competence contributions of cases it is necessary to model the interactions between cases in terms of how their coverage and reachability sets overlap.

Def. 3 RelatedSet(c)=CoverageSet(c)∪ReachabilitySet(c)

Def. 4 For c1, c2 ∈ C, SharedCoverage(c1, c2)
iff [RelatedSet(c1) ∩ RelatedSet(c2)]≠{ }

Def. 5: For G = {c1,...,cn} ⊆ C, CompetenceGroup(G)
iff ∀ci∈G,∃cj∈G- {ci}: SharedCoverage(ci,cj) ∧
∀cj∈C-G,¬∃cl∈G: SharedCoverage(cj,cl)

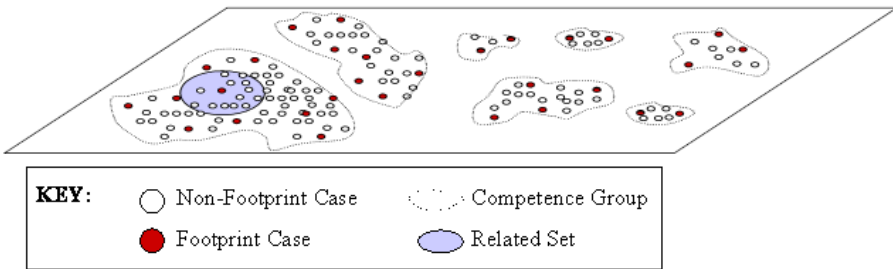


Fig. 1. A sample case-base showing competence groups, footprint cases, and related sets

First we define the related set of a case to be the union of its coverage and reachability sets (see Def. 3). When the related sets of two cases overlap we say that they exhibit *shared coverage* (see Def. 4) and cases can be grouped together into so-called *competence groups* which are maximal sets of cases exhibiting shared coverage (see Def. 5). In fact, every case-base can be organised into a unique set of competence groups which, by definition, do not interact from a competence viewpoint – that is, while each case within a given competence group must share coverage with at least one other case in that group, no case from one group can share coverage with any case from another group (see Figure 1). Moreover, this vital property of competence groups means that each group makes an independent contribution to overall

competence. In fact we argue that it is the competence groups, not the individual cases, that are the *fundamental units of competence* in case-based reasoners.

2.3 Footprint Cases

While every competence group makes a unique (and independent) contribution to competence, not every case in a competence group makes a positive competence contribution; for example, Smyth & Keane [11] have shown that so-called *auxiliary* cases make no competence contributions. The *footprint cases* of a competence group are those cases that do make a positive competence contribution and the *footprint set* is that minimal set of group footprint cases that collectively provides the same coverage as the entire group. The footprint set is important because it is only these cases that we need to consider when estimating the competence properties of a given group, the non-footprint cases are irrelevant from a competence viewpoint (although they may be relevant from an efficiency viewpoint) – see also Figure 1.

```

G, Competence Group
COV-FP (G)
R-Set ← cases in G; FP ← {}
While R-Set is not size empty
    C ← case in R-Set with largest coverage set size
    FP ← FP ∪ C
    R-Set ← R-Set - CoverageSet(C)
    Update coverage sets of cases in R-Set
EndWhile
Return (FP)

```

Algorithm 1. Computing the footprint set of a competence group

Algorithm 1 is one of many algorithms that we have explored to compute footprint sets (see also [12,13,14]). The algorithm adds cases with the largest coverage sets to the growing footprint set (FP). Each time a case is added, all of the cases that it covers are removed from the remaining case set (R-Set).

2.4 Relative Coverage & Global Competence

Each footprint case is chosen because it makes a positive contribution to group competence, but this does not mean that each footprint case makes the same competence contribution. In fact, one of the insights of the work of Smyth & Keane [11] is to demonstrate that cases in a CBR system will tend to vary greatly in their competence contributions. In order to compute the actual competence contribution of a footprint case we need to estimate its competence relative to other group cases, and this is the role of the *relative coverage* estimate (see Def. 6). It is based on the idea that if a case c' is covered by n other cases then each of the n cases will receive a contribution of $1/n$ from c' to their relative coverage measures.

Def 6:
$$\text{RelativeCoverage}(c) = \sum_{c' \in \text{CoverageSet}(c)} \frac{1}{|\text{ReachabilitySet}(c')|}$$

The competence contribution of a competence group then is simply the sum of the relative coverage values of the group's footprint cases. In turn, since competence groups make independent competence contributions, the competence of the case-base as a whole is simply the sum of the group competence estimates.

This completes the outline description of our competence model. With this model we can estimate and predict the competence of different case-bases. Moreover, it is now possible to assess the competence contributions of individual cases, a vital component in any case addition/deletion maintenance strategy.

3 An Efficient Procedure for Updating the Competence Model

Each time a case is learned by a case-based reasoner the competence model must be updated and this contributes an additional cost to the learning process. In this section we explain how to reduce this cost by describing an efficient model-update procedure.

3.1 The Standard Update Procedure

Before describing this new update procedure it is useful to begin with a description of what might be termed the standard update procedure (shown in Algorithm 2). This algorithm consists of three basic steps. First, the local competence characteristics, the coverage and reachability sets (and therefore the related sets), of the new case must be computed. This involves comparing every case in the case-base to the newly learned case in order to determine whether these cases can solve, or be solved by, this new case; this process is $O(n)$ in the size of the case-base.

Second, the competence group membership of the new case must be computed and there are a number of conditions that must be checked during this computation. If the case cannot be solved by, or cannot itself solve, any other case in the case-base – such a case is termed a pivotal case by Smyth & Keane [8,11] and will have a related set that contains only itself – then it will belong to a new singleton competence group. However, usually the new case will be solved by, or can solve, at least one other case and will therefore have a related set containing a number of other cases. If all of these cases belong to the same competence group then the new case will also belong to this group. However, if these cases belong to different groups then the new case is a so-called spanning case (see [8,11]) and results in the merging of these competence groups into one new competence group. This new group is made up of the union of the cases from the merged groups plus the new case.

The third step involves updating the footprint set of the competence group containing the new case. Strictly speaking this should involve recomputing the footprint set for the affected group but a more efficient procedure is available. If the new case is a pivotal case and resulted in the creation of a new singleton competence group then this new group will have a footprint set that contains just this pivotal case.

If the new case was found to be part of one existing competence group then it can only impact on the footprint set of this group if it is not currently covered by the cases in this footprint set; that is, if the intersection between the footprint set and the new case's reachability set is the empty set. If this is true then the new case is added to the group footprint set. Finally, if the new case is a spanning case, and has resulted in the creation of a new group by merging existing groups, then the footprint set of this new group is the union of the footprint sets of the merged groups (for simplicity this is carried out as part of the group membership update in Algorithm 2) and if the new case is not covered by any cases in this footprint set then it is added.

```

C - learned case, CB - case-base
Model-Update(C, CB)

  1. Compute Local Competences
  For each case  $x \in CB$ 
    If Solves(C, x) then
      Add x to CoverageSet(C)
      Add C to ReachabilitySet(x)
    EndIf
    If Solves(x, C) then
      Add x to ReachabilitySet(C)
      Add C to CoverageSet(x)
    Endif
  EndFor

  2. Update Group Membership
  If RelatedSet(C) - {C} = {} create new group G containing C
  Else
    If all cases in RelatedSet(C) belong to same group, G
      Add C to G
    Else create a new group, G, by merging the groups
      that the cases in RelatedSet(C) belong to.
      FootprintSet(G) = union of footprint sets of
        Merged groups.
  EndIf

  3. Update Group Footprint
  If RelatedSet(C) - {C} = {} then FootprintSet(G) = {C}
  Else if FootprintSet(G)  $\cap$  ReachabilitySet(C) = {} then
    Add C to FootprintSet(G)
  EndIf

```

Algorithm 2. The Standard Competence Model Update Procedure

3.2 A More Efficient Update Procedure

The majority of the cost of the standard update procedure is associated with the calculation of the local competence characteristics of the new case; $O(n)$ in the size of the case-base. By comparison the group membership and footprint update costs are very low. Therefore, any efficiency gains in the local competence calculation will have a significant impact on the overall cost of updating the model. In this section we describe a more efficient competence model update procedure (see Algorithm 3) that

benefits from significant reductions in the cost of computing the related set for the newly learned case. The basic idea is to compute the related set for the new case by examining only a small fraction of the cases in the case-base. This makes sense because in reality only a small fraction of the cases in the case-base will find their way into the coverage and reachability sets of a new case – most cases in the case-base cannot solve, or be solved by, a newly learned case.

```

C - learned case, CB - case-base, k
Compute Local Competences(C, CB, k)
  Ordered-FS ← FootprintSet(CB) sorted in descending
    order of similarity to C
  Footprint-Cases ← first k cases in Ordered-FS
  Update-Set ← Union of related sets of Footprint-Cases
  For each case x ∈ Update-Set
    If Solves(C, x) then
      Add x to CoverageSet(C)
      Add C to ReachabilitySet(x)
    EndIf
    If Solves(x, C) then
      Add x to ReachabilitySet(C)
      Add C to CoverageSet(x)
    Endif
  EndFor

```

Algorithm 3. A more efficient procedure for computing the local competence characteristics of a newly learned case

Of course, if the related set is to be computed with respect to a small subset of the case-base then it is critical that this subset (which we will call the *update set*) is carefully selected so that it contains only those cases that are likely to be members of the new case's related set. If any related cases are missing from the update set then they will never be added to the new case's related set, resulting in model errors.

We propose a two-step method for computing the update set for a new case, which can be adjusted in favour of update efficiency or accuracy as required. The first step of this method is to compare the new case to each of the cases in the footprint set of the entire case-base; that is, the union of each of the competence group footprint sets. This footprint set is sorted in ascending order of similarity between its cases and the newly learned case to produce an *ordered footprint set* (Ordered-FS in Algorithm 3). Step two then computes the update set as the union of the related sets of the first k footprint cases in the ordered footprint set. The coverage and reachability sets for the new case can now be determined in the usual way but with reference to the update set only, rather than the entire case-base.

This procedure has the potential to significantly reduce the cost of computing the new case's related set since the update set will contain a small fraction of the cases in the case-base. The cost of computing the update set, which involves comparing the new case to each case in the case-base footprint set, also remains low since the size of the footprint set is generally a fraction of the entire case-base. The accuracy of the update set can be tailored by adjusting k, the number of footprint cases that need to be examined. Low values of k improve efficiency by producing small update sets, but

these sets may be missing relevant cases. Higher values of k increase the likelihood that the update set will contain all of the relevant cases, but efficiency is reduced.

4 Experimental Analysis

In the previous sections we have outlined a proven model of competence for case-based reasoners, and introduced a new procedure for updating this model to accommodate the run-time learning of new cases. We claim that this new update procedure will offer significant cost savings without loss of model accuracy, and in this section we back up these claims with empirical evidence.

4.1 Experimental Setup

The test data for our analysis comes from two freely available case-bases. The Travel case-base (available from the case-base archive on the AI-CBR web-site at <http://www.ai-cbr.org>) contains 1400 cases from the travel domain. Each case describes a package holiday in terms of features such as location, style, accommodation, number of people, price etc. The Property case-base (available from the UCI Machine Learning Repository, [1]) contains cases from the residential property domain, each describing an individual residential property in terms of features such as, location, style, facilities, crime rates, price, etc.

We built CBR systems for these case-bases. For each system the solvability criterion was based on a similarity threshold; a target problem was successfully solved if the similarity between it and the retrieved case exceeded the threshold.

4.2 Update Efficiency

In this experiment we compare the cost of competence model updates using the standard and new update procedures. Moreover, we focus on the local competence update costs since this is where the two procedures differ and, in the standard procedure, this accounts for the lion's share of the update costs.

As explained in Section 3, the cost of computing the local competence characteristics by using the standard update procedure is $O(n)$ in the size of the case-base; that is, using our experimental CBR systems, each time a new case is added to the current case-base, it must be compared to every case in this case-base to determine which cases can solve it and which cases it can solve. In contrast, the corresponding cost in the new update procedure is based on p comparisons to compute the update set (that is, each of the p footprint cases must be compared to the new case) plus q comparisons, since each of the q cases in the update set must be compared to the new case to assess solvability. In this experiment we measure these costs (n and $p+q$) during learning for the Travel and Property systems.

Method: The Travel and Property systems are initialized to contain 400 and 200 cases respectively and the competence model is built for each case-base from scratch. Next we emulate run-time case learning by adding the remaining cases to each case-

base and updating the competence models accordingly. In total, four competence models are maintained for each system, one based on the standard update procedure and three based on the new update procedure with values of k of 1, 3, and 5. We compute the update cost (in terms of the number of case comparisons during the local competence computation) for each case update and calculate the speed-up ratio $(n/p+q)$, for each value of k . To remove any ordering effects from the results we repeat this process 100 times each time using a different random ordering of the Travel and Property case-bases – this guarantees different initial case-bases and ensures that the order in which cases are learned varies with each run. This produces 100 different speed-up ratios for each update from which a mean value is computed.

Results: The results are shown in Figure 2(a&b) as graphs of mean speed-up versus case-base size for the Travel and Property domains respectively. Each graph consists of three plots and each plot corresponds to the speed-up profile produced for a given value of k (1, 3 or 5, as indicated).

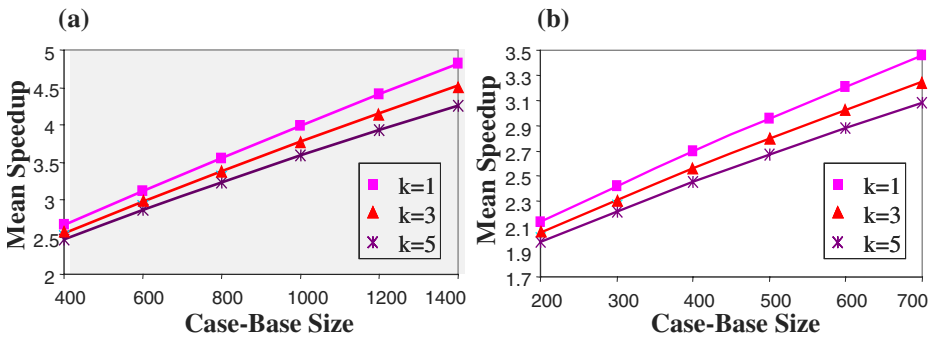


Fig. 2. The speed-up of the new model update procedure compared to the standard update procedure for the (a) Travel and (b) Property domains

Discussion: The results indicate that there are cost savings associated with the new update procedure. For example, in the Travel domain, Figure 2(a), we attain a speed-up value of nearly 5 for the model update associated with a 1400 case case-base (at the $k=1$ setting). This speed-up value indicates that the new procedure examines only 20% of the cases that the standard procedure must examine. We also see that the potential speed-up is increasing with case-base size. This means that the cost saving associated with the new update procedure is increasing, relative to the standard cost, as the case-base grows. Finally, as expected, as the value of k increases we witness a small decrease in speed-up; for example we note a 2.5% decrease in speed-up for each single increment of k . These results are found to be consistent in both domains and in summary we can conclude that the new update procedure offers significant and reliable cost savings over the standard procedure.

4.3 Model Accuracy

The efficiency benefits of the update procedure are meaningless if model accuracy is compromised; key cases could be missing from the coverage or reachability sets computed by the new procedure. We look at this issue here.

Def 7.
$$\text{Accuracy}(\text{Set1}, \text{Set2}) = 100 \cdot \frac{|\text{Set1} \cap \text{Set2}|}{|\text{Set1} \cup \text{Set2}|}$$

Method: We follow the same method used in the previous experiment. However, after each update, instead of noting the update cost, we use the formula in Def. 7 to measure the percentage accuracy of the coverage and reachability sets produced by the new update procedure with respect to the equivalent sets produced by the standard update procedure. For each update this gives a coverage set accuracy value and a reachability set accuracy value and these are combined to produce a mean accuracy value for that update. This results in 100 mean accuracy values for each update to a case-base of size n over the 100 different case orderings. The 100 values are combined to produce an overall mean accuracy for each case-base size.

Results: The results are shown in Figure 3(a&b) as graphs of overall mean accuracy versus case-base size for the Travel and Property domains respectively. Each graph consists of three plots and each plot corresponds to the accuracy profile of the new update procedure for a given value of k as indicated.

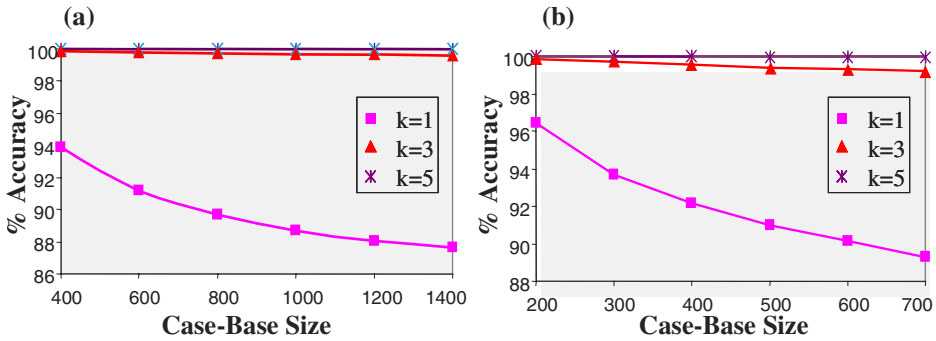


Fig. 3. The percentage accuracy of the competence models produced by the new and standard model update procedures for the (a) Travel and (b) Property domains

Discussion: The results are very positive. As expected, the accuracy of the model improves with increasing values of k. In both domains the new update procedure is seen to deliver 100% accuracy at k=5. This level of accuracy drops for lower values of k. For example, for a 1400 case Travel case-base, at k=3 the accuracy is 99.6%, which means that only 1 case in 250 is missing from the local competence sets produced by the new update procedure. In both domains, there are significant accuracy jumps between the values of k=1 and k=3. Accuracy is also seen to slowly degrade as the case-base grows in size; this is especially noticeable for low values of k (see k=1 plots in Figure 3(a&b)). This is expected as update errors have a cumulative effect, and any errors that creep in early on can result in the introduction of additional errors in the future. This cumulative effect is small, and reduces for increasing values of k. In the Travel domain, the accuracy drops from 99.9% (at the 400 case level) to 99.65 (at the 1400 case level) for the k=3 setting in the new update procedure, a fall of only 0.25% in accuracy; that is, less than 4 new errors in 1000 case updates. A similar pattern of results is found in the Property domain. Thus, the new update procedure provides significant efficiency benefits without compromising

model accuracy. In fact, the new procedure manages to attain perfect accuracy levels at reasonable values of k , for which significant efficiency gains are still available.

5 Discussion

The story so far shows that our competence model update procedure can significantly reduce the cost of maintaining the competence model without compromising its accuracy or effectiveness. There is one final twist in the tale that deserves discussion.

In previous work we have described how the competence model can be used to drive a novel case retrieval technique (called *footprint-based retrieval*) that can deliver superior retrieval results by examining only a fraction of the cases in the case-base [14]. The interesting feature of this retrieval technique is that the competence model is used to select a subset of cases (called the *retrieval set*) for examination during retrieval. Moreover, the method used for selecting this subset of cases, and the subsequent comparisons between the target problem and this subset, is very similar to the method used by our new model update procedure to compute the local competence characteristics of a newly learned case. In other words, the same basic computations are carried out in footprint-based retrieval and model update – the target case in retrieval corresponds to the learned case in model update, the selection of the retrieval set corresponds to the selection of the update set, and the retrieval comparisons correspond to the local competence comparisons.

Therefore, in a CBR system using footprint-based retrieval, the cost of updating the competence model can come almost for free, since the task of computing the coverage and reachability sets for the newly learned case is carried out as a normal part of the footprint-based retrieval that led to this new case. In other words, starting with a target problem, the footprint-based retrieval procedure selects the most similar case in the case-base, this case is adapted and (assuming learning is appropriate) the target problem description plus the newly adapted solution are packaged as a new case and added to the case-base. Finally, if the similarity between the target and a case can be used to estimate the solvability of the target with respect to the case, then the computations performed during retrieval to calculate the similarity between the target and its related cases in the case-base, can be reused to determine the coverage and reachability sets of the new case during the competence model update.

6 Conclusions

We have outlined an effective competence model for CBR systems and introduced a new procedure for updating this model at run-time. We have also provided empirical evidence to support claims that this new update procedure offers significant efficiency benefits over the standard update method without compromising model accuracy. We believe that explicit models of performance (such as competence and efficiency models) are of fundamental importance to the CBR community, as we believe that such models may hold the key to a range of important CBR problems. For instance, we have already shown how our competence model can be used to develop novel case

addition, retrieval, case-base visualisation, and authoring support solutions [12,13,14]. Our new update procedure means that these solutions are available at a reduced cost.

References

1. Blake, C., Keogh, E. & Merz, C.J: UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science (1998)
2. Francis, A.G. and Ram, A.: A Comparative Utility Analysis of Case-Based Reasoning and Control Rule Problem Solving. In: Proceedings of the 8th European Conference on Machine Learning (1995)
3. Kolodner, J. Case-Based Reasoning. Morgan-Kaufmann, San Mateo, California (1993)
4. Leake, DB. Case-Based Reasoning: Experiences, Lessons, and Future Directions. MIT Press (1996)
5. Leake, DB, Wilson DC: Categorizing Case-Base Maintenance: Dimensions and Directions. In: B. Smyth & P. Cunningham (Eds.) Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, LNAI 1488. Springer-Verlag (1998). 196-207
6. Minton, S.: Qualitative results concerning the utility of explanation based learning. Artificial Intelligence, **42**(2-3) (1991) 393-391
7. Racine, K. & Yang, Q: Maintaining Unstructured Case Bases. In: DB Leake & E. Plaza (eds.) Case-Based Reasoning Research and Development. Lecture Notes in Artificial Intelligence, LNAI 1266. Springer-Verlag (1997) 553-564
8. Smyth, B.: Case-Base Maintenance. In: AP del Pobil, J. Mira, M. Ali (eds.) Tasks and Methods in Applied Artificial Intelligence, LNAI 1416. Springer-Verlag (1998). 507-516
9. Smyth, B. & Cunningham, P.: The Utility Problem Analysed: A Case-Based Reasoning Perspective. In: I. Smith, & B. Faltings (eds.): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 1168. Springer-Verlag (1996) 392-399
10. Smyth, B. & Keane, M.T.: Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning. Artificial Intelligence, **102** . (1998) 249-293
11. Smyth, B. & Keane, M.T.: Remembering to Forget: A Competence Preserving Deletion Policy for Case-Based Reasoning Systems. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. Morgan-Kaufmann. (1995) 377-382
12. Smyth, B. & McKenna, E. Modelling the Competence of Case-Bases. In: B. Smyth & P. Cunningham (eds.) Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, LNAI 1488. Springer-Verlag, (1998). 208-220
13. Smyth, B. & McKenna, E. Building Compact Competent Case-Bases. In: K-D Althoff, R. Bergmann, L-K Branting (eds.) Case-Based Reasoning Research & Development. Lecture Notes in Artificial Intelligence, LNAI 1650. Springer-Verlag, (1999) 329-342
14. Smyth, B. & McKenna, E. Footprint-Based Retrieval. . In: K-D Althoff, R. Bergmann, L-K Branting (eds.) Case-Based Reasoning Research & Development. Lecture Notes in Artificial Intelligence, LNAI 1650. Springer-Verlag (1999) 343-357
15. Zhu, J. & Yang, Q: Remembering to Add: Competence-preserving Case Addition Policies for Case Base Maintenance. In: Proceedings of International Joint Conference in Artificial Intelligence (IJCAI-99). Morgan-Kaufmann. 234-241