# Knowledge Maintenance in myCBR

Vani Aul, Christian Severin Sauer, Estiak Bulbul, David C. Wilson, and Thomas Roth-Berghofer

School of Computing and Technology, University of West London, UK
vani.aul,christian.sauer,thomas.roth-berghofer@uwl.ac.uk,davils@uncc.edu

**Abstract.** CBR systems, being knowledge based systems, process knowledge. Due to changes in the environment a CBR system's knowledge model can become outdated, thus creating a need for constant maintenance of said knowledge model. In this paper, we describe an implementation of (semi-)automatic knowledge maintenance of two of the four knowledge containers of CBR systems, specifically case base maintenance and maintenance of similarity measures within the CBR system development SDK *myCBR*. We describe our approach to create, elicit and manage quality measures that are used to trigger maintenance actions if the quality measures fall below defined thresholds, indicating a declining efficiency/accuracy of a case base or particular similarity measure. We further detail on the implementation of our approach into *myCBR Workbench* to enable a knowledge engineer to incorporate the notion of maintenance already at the design stage of a CBR system. The approach relies on the notion of maintenance attributes to be able to measure the quality of case bases and similarity measures. Initial experiments using the newly introduced quality measurement attributes indicate that our approach is promising.

**Keywords:** Knowledge maintenance, Case base maintenance, Usage knowledge acquisition

## 1 Introduction

A CBR system represents its problem solving experience in the form of cases. In the process of problem solving appropriate cases, i.e., cases most similar to a problem case, are presented to the users as suggested solutions. The suggested cases should be relevant, correct and capable of dealing with the user's current problem scenario. The suggested cases must be competent enough to solve the user's problem. In order to make sure that the CBR system stays competent and efficient even in an environment of contextual and other changes, maintenance of the CBR system's cases, being represented in its case base, should be carried out at regular intervals.

Maintaining the knowledge of any knowledge-based systems is vital for its continued use [8]. CBR systems, being knowledge based systems, process knowledge. Although such systems do not *wear out* there are still changes in the environment of these systems that do affect their functionality. Their knowledge

model can become outdated by the on-going changes in its environment, thus creating a need for the constant maintenance of said knowledge model. The intensity of changes of different variables in the environment varies in different CBR systems depending on the nature of the domain knowledge represented in a specific CBR system. Additionally to the necessity of maintaining the knowledge model in a dynamic environment, knowledge maintenance is also often able to increase the performance of knowledge models.

Next to the cases a CBR system's knowledge is stored in three further knowledge containers [5,6]. The knowledge in the vocabulary, similarity measures and adaptation knowledge containers is also in need of being maintained to provide an efficient and up-to-date CBR system. In the context of these demands our current research is focused on knowledge maintenance for CBR systems to maintain their competence, effectiveness and efficiency with accurate, relevant and complete knowledge. To enforce a regular, at least semi-automatic, maintenance interval we propose that the maintenance tasks in a CBR system can be carried out by employing a control loop, to observe the CBR system's performance, and to counteract any change that is not desired.

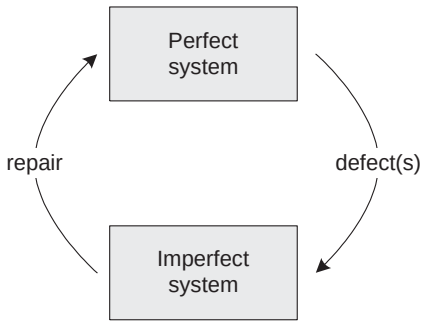


**Fig. 1.** Control loop [7]

The triggering of this loop can be time driven, which ensures a regular maintenance but has the disadvantage of probably running unnecessary maintenance tasks. A different approach to triggering the maintenance tasks is to monitor the system's overall performance as well as competence and start the maintenance tasks if certain quality measures for the system's performance are breached, indicating a deteriorating quality of the system's problem solving abilities. In order to integrate such a control loop in a CBR system, the most commonly used CBR process model of Aamodt and Plaza [1] can be enhanced with two more steps: *Review* and *Restore* [4]. The *Review* step observes the system during the retrieval step and assesses the quality of, for example, retrieved cases. The *Restore* step becomes effective if the review step reveals a deteriorating case quality and invokes a set of maintenance steps that *repair* the CBR system's knowledge model to get back to a desired level of quality. These two new steps *Review* and

*Restore* are the main focus of this paper along with their implementation on the *myCBR* SDK.

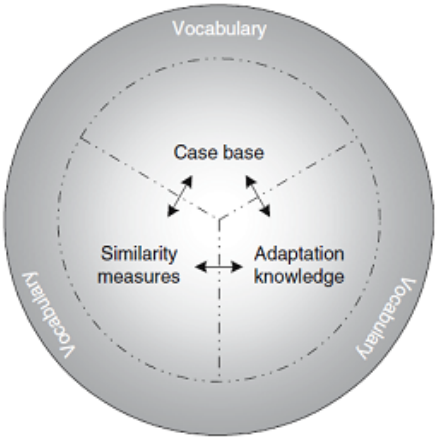


**Fig. 2.** The four knowledge containers of CBR [5,6]

As shown in Figure 2, the four knowledge containers of a CBR system hold the available knowledge of a CBR system [5,6]. The vocabulary (classes, attributes, predicates etc.) comprises the knowledge model. The similarity measures are used to compare cases with queries. The adaptation knowledge adapts past solutions to current problems. The case base stores the cases.

The rest of this paper is structured as follows: In section 2 we provide an insight in the motivation of our work. In section 3 we interlink our work with related work on CBR and knowledge maintenance. We, then, introduce our approach to knowledge maintenance in section 4. Section 5 demonstrates the use of maintenance attributes to gather and store case usage data that is then employed to indicate necessary maintenance actions, should the need for such actions arise. In section 6 we then introduce our aim to further broaden our initial approach to be employed on similarity measures and generic feature maintenance at all. We document our testing and evaluation of our initial approach in section 7 and provide our conclusions and planned future work in section 8.

## 2 Motivation

The knowledge contained in the case base needs to be accurate and up to date with contextual changes in the environment and also competent to serve the user's requirements. A reliable case base in a CBR system can be achieved by carrying out its maintenance on a regular, or, even more desirable, on an automated 'on demand' basis. To enable such an automated approach to triggering

case base maintenance tasks on demand, we introduce our approach of tracking the usage-data of the CBR system. Based on this usage-data tracking we track the number of retrievals, count the frequency of appearance of a case in the retrievals, shortlist the top and least performing/retrieved cases, study potential patterns of top or low performing cases, establish case-attribute's levels of influence and use all this data to define and establish quality measures for the performance, accuracy and competence of our case base. We then further use these quality-measures and experimentally established thresholds for these quality measures to automatically trigger case base maintenance tasks when these thresholds are breached, indicating a deteriorating quality of the case base. By following this approach we are able to maintain the efficiency of a CBR system with continuous refinement and maintenance of its knowledge model.

Our approach of gathering case usage data also limits our approach, for now, to CBR systems which show a high case cohesion and thus exhibit a high regularity in their cases as well as a direct re-use of the solutions retrieved [2]. Maintenance of a CBR system has to be a key element in any CBR system already during its design phase [7], thus allowing for the building in of maintenance abilities as well as, with our approach, also building in observation abilities with regard to the system's case base's performance. Our approach to such an integration of performance observing features in the CBR system during its design, for the kind of CBR systems outlined, integrates maintenance and observation features into the *myCBR*SDK. Although we implemented our approach already and performed initial experiments with our implementation we are eager to get feedback in this early stage of our approach to direct and refine our future work.

## 3 Related Work

Changes in the environment affect the accuracy and competence of a CBR system's knowledge model and thus its functionality [8]. Following this view it can be stated that the maintenance of CBR systems may not only include incorporating new knowledge but also updating or modifying already existing knowledge for making the CBR system better.

It is important to keep a check on the case base's size as well as to detect the incorrect or inconsistent cases. A maintenance approach introduced by Smyth and McKenna is based on the performance model of a CBR system [9]. As the case base is the main source of competence of a CBR system, each case contributes positively or negatively to this competency. It can be possible to measure if the case is contributing positively or negatively by calculating the coverage versus the reachability of an individual case in the case base. Coverage describes the kind of target problems a case is able to solve, whereas reachability describes the set of cases that can be used for solving a specified target problem. Cases with high coverage seem to be making a large contribution to the system's competency, in contrast to cases with a high reachability. High reachability means that many cases exist that can solve similar problems. This maintenance approach relies mainly on deleting incompetent cases. Additionally,

as we already stated in section 2, our approach assumes a CBR System with a high case cohesion, introduced by Lamontage [2].

Leake and Wilson [3] highlight the importance of conducting case base maintenance by balancing the competence-performance dichotomy in the light of essential goals and constraints of a CBR system. Leake and Wilson state the idea that it is essential for a CBR system to meet the top level goals namely:

- Problem solving efficiency (average time to solve a problem)
- Competence (range/number of target problems solved)
- Quality of solutions in solving problems (level of errors in solutions)

In addition to the above quality goals Leake and Wilson suggest that case base maintenance should be guided by important constraints including size limits of case base, long and short term performance goals in expected future problems. Leake and Wilson conducted empirical experiments and discovered that if the problem distribution within the case base is non-uniform, case base maintenance needs to be guided by performance rather than compactness and competence only. If compactness is focused on as a main goal for maintenance then it might miss opportunities to increase efficiency.

From the above we can see that CBR maintenance is definitely a pre-requisite for effective CBR systems. This pre-requisite can be achieved by knowledge maintenance in cases to keep information updated in cases and by cases deletion from case base to increase the compactness and efficiency of case bases. Based on the approaches described above, we have begun to create a maintenance perspective in *myCBR Workbench* for tracking performance of cases and based on this tracking, informing the maintenance engineer, so she can conduct the necessary maintenance actions should the need arise.

## 4 Our Approach

Our main objective is to create a maintenance perspective in *myCBR* to assess the data on case usage. Case usage data allows the generation of quality measures which can then be used to trigger the maintenance of the knowledge in the case base, vocabulary, and similarity measures. Currently we have developed a prototype version of *myCBR Workbench* to demonstrate our approach.

To allow for this new maintenance perspective, we extended *myCBR* to generate usage-data when individual cases are accessed during retrieval. Based on manual evaluation we further defined threshold values for quality measures to monitor the top performing / most retrieved and least performing / least retrieved cases. We also studied the common patterns or values in the top performing cases as opposed to the least performing cases. We allow for adjusting these values in the least performing cases and then observe the performance impact resulting from these changes. In case the performance deteriorates we provide the necessary features to reverse these changes. By automating the measurement of the quality measures and subsequent triggering of the maintenance tasks we aim for a persistent control loop to manage the maintenance of the case

base. The automated control loop we introduced can, as already stated, be seen as two further sub-processes of the retrieval process in the CBR cycle.

Following Leake and Wilson [3], the main objective is to maintain optimal performance and competence within our case bases via maintenance. To enhance the performance, we have aligned our approach to the principle of *'Observe Performance to Improve Performance'*. As shown in Figure 1 we follow an iterative process to remove the defects from an imperfect system, thus constantly improving the CBR system's knowledge model.

## 5 Maintenance Attributes and Basic Measures

The current structure of a case in *myCBR* contains attributes holding the information defining the actual case. In addition to these *case-building* attributes, we introduce *maintenance attributes*. These attributes are implemented on the *myCBR* SDK and, unlike the case-building attributes are not used for retrieval, i.e., problem solving. They store performance information on a case. This performance information consists of retrieval counters and other measures for operations performed in which the particular case was involved. The values stored in the introduced *maintenance attributes* are used together with defined threshold values to indicate a case's performance and highlight the need for maintenance of the case, should it occur. The new *maintenance attributes* are:

- *Cancelled*: Total number of rejections of a retrieved case by the user
- *HitRate*: A ratio of *Total Retrieved* divided by *Total Liked*
- *LastFeedback*: Free text feedback on the case
- *LastModified*: Date of the last modification of the case
- *LastRetrievedDate*: Date of the last retrieval of the case
- *NoOfModification*: How often were modifications applied to the case?
- *RetrievalAfterModified*: Number of Retrievals since last modification
- *TotalLiked*: Number of times a retrieved case was deemed useful by a user
- *TotalRetrieved*: Total number of times a case was retrieved in its lifetime
- *TotalSelected*: Total number of times a case was accepted or 'bought' by a user

We chose the above listed maintenance attributes to cover the temporal aspect of a case, being new or old, the frequency of a case being retrieved, the quality of the solution the case offers as well as the maintenance effort that already was invested in the case. So the chosen maintenance attributes provide key information on individual case's performance in the case base. We established, as introduced, threshold values for the values of the *maintenance attributes*. Next to this the *maintenance attributes* values get updated in the on-going use of the CBR system, for example, the *Total Retrieved* attribute value increases by 1 if the case is amongst the top 5 retrieved cases.

## 6 Implementation of maintenance features in *myCBR*

To allow for a maintenance perspective we introduced our approach of tracking the performance and usage of individual cases by using *maintenance attributes*. Additionally, we also intend to update the *myCBR* SDK in a way that it generates case base usage logs. A combination of the *maintenance attributes* data and case base usage logs' data would inform us on the pattern of retrieval, usage and performance of individual cases. Additionally we would be informed about the case's usage in the context of different case base(s) used within a project.

Additional to the SDK extension we aim to add a new maintenance perspective to the GUI of the myCBR workbench. The goal of this new perspective will be to present all available usage data to the knowledge engineer while also allowing her to manually update or correct this data. The usage data gathering, for now, is realised partly by introducing new ways into *myCBR*to gather user feedback directly by providing buttons for the user to "'Like, Unlike or Select(Buy"') a retrieved (recommended) Holiday Package case. Figure 4 shows these newly integrated buttons in the retrieval result view of *myCBR*. A later stage of the implementation of our approach will provide these functionalities of gathering user feedback within the SDK to allow for their integration into the front-ends of applications employing a CBR system developed with *myCBR*.

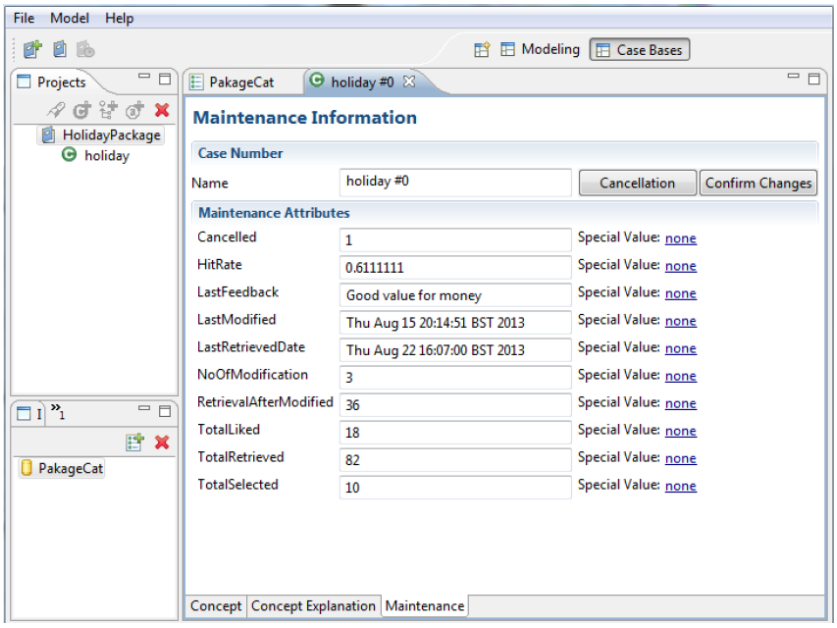


**Fig. 3.** The newly added maintenance view in *myCBR*

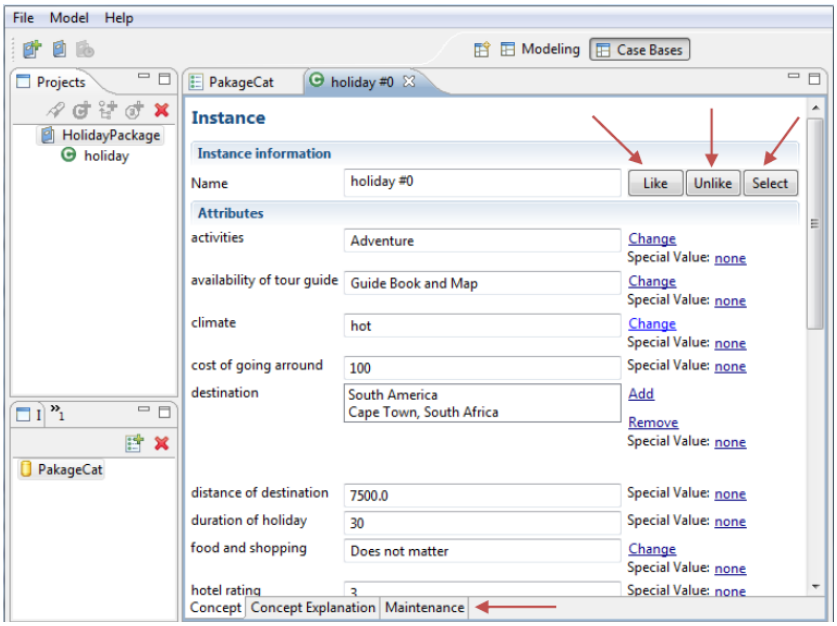


**Fig. 4.** The user feedback buttons in the retrieval result view in *myCBR*'s Case Bases perspective

At the current time we implemented the maintenance attributes temporarily, simulating these attributes in *myCBR* by labelling them in a specific way. Our next step is now to integrate the maintenance attributes into *myCBR* as a new class of attributes to make them available in a separate maintenance perspective which will be added to the two existing perspectives, knowledge modelling and case base management. Thus we integrate the maintenance functionally into the SDK, to allow it to be used by applications employing a CBR system created using *myCBR* and into the GUI of *myCBR Workbench* to allow for the integration of maintenance functionalities into a CBR system being developed using *myCBR*. Figure 3 shows the maintenance view in its current state as well as the maintenance attributes implemented for the mock-up holiday package recommender system we employed within our experiments.

To further refine the ability to gather usage data, building the base for evidence-based maintenance assessment, we aim to establish also a new set of attributes to provide more tailored maintenance information on the similarity measures. We therefore plan to extend the classes implementing the attributes of a case within *myCBR*. With these future maintenance functionalities implemented we then can, if cases are not the reason for a declining performance/quality of a CBR system, inspect the similarity measures which then are most likely faulty. An example for a characteristic 'misbehaviour' of a CBR system based on faulty similarity measures is the generation of "'clusters"' of cases which are repeatedly

retrieved despite many other possible similar cases (generation of favourite sets). These symptoms are strong indicators of insufficient modelling of the similarity measures.

## 7 Case study

To test our initial partial implementation of case maintenance in *myCBR* we developed and deployed a test case base of fictitious Holiday Packages to be recommended to potential customers. The newly implemented maintenance information then was generated to identify obsolete case information, top performing, or negatively performing cases. The maintenance information generated was then used to inform the knowledge engineer of necessary maintenance actions, should they become necessary based on the maintenance information gathered. The use of the maintenance information to evaluate the performance of the recommender system and amend the case base based on this information resulted in a competent CBR system with good accuracy and performance.

Our procedure of maintenance was structured as follows:

- Formulate a set of performance standards for the Holiday Packages (HP) cases
- Assess performance of the HP cases
- Define a mechanism / criteria for accessing data accumulated in the maintenance attributes
- Identify and analyse the attributes or cases requiring maintenance, based on the maintenance information gathered
- Apply the necessary maintenance changes to update the cases and attributes
- Evaluate the effectiveness of the applied maintenance measures

For our experiment with the mock up Holiday Package recommender systems seven test-users were selected to query the system for Holiday package recommendations. The experiment then was parted into two different stages representing two iterations of the newly introduced maintenance loop, using the newly implemented maintenance information stored in the new maintenance attributes.

The goal at the end of each iteration was to find neglected/none retrieved cases, then find out what qualities they were missing based on the accumulated maintenance data. 70 retrievals were carried out in each phase to emulate the usage of the recommender systems and allow for the accumulation of case usage information within the maintenance attributes. For each of the first 70 random queries of the first phase we asked the test-users and for their feedback on the quality and accuracy of the Holiday Package cases the system recommended to them. We did so in our initial test with the initial case base to gather the case usage data we needed as input for the second phase of our experiment.

After the retrievals of the first phase that accumulated case usage information we then used this data to maintain, optimise the initial case base. We then again asked the 7 test-users to again do 10 retrievals and asked their feedback on the accuracy, quality of these retrievals. Evaluating the feedback on the second set

of 70 retrievals to the, now refined, case-base, we were able to establish a definite increase in the quality of the retrievals. We also noticed a reduced distribution of the times the top-rated and the least-top rated case were retrieved, indicating a more even 'usefulness' of the remaining cases in the case base.

## 8 Conclusion and future work

In this paper we have introduced our approach to gather case usage data with the help of newly introduced maintenance attributes within our CBR development software *myCBR*. We have done so to support the knowledge maintenance of two knowledge containers, case base and similarity measures. We have described our implementation of maintenance attributes on the *myCBR* SDK.

As our immediate future work we will further enhance the number of quality measures on the *myCBR* SDK to allow for usage data functionalities to be available in any CBR system using the *myCBR* SDK via *myCBR*'s API.

## References


1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 1(7) (Mar 1994), `ftp://ftp.ifi.ntnu.no/pub/Publikasjoner/vitenskaplige-artikler/aicom-94.pdf`; letzte Verifikation 11. Juni 2007
2. Lamontagne, L.: Textual cbr authoring using case cohesion. In: Proceedings of 3rd Textual Case-Based Reasoning Workshop at the 8th European Conf. on CBR. pp. 33–43. Citeseer (2006)
3. Leake, D.B., Wilson, D.C.: Categorizing Case-Base Maintenance: Dimensions and Directions. In: Smyth, B., Cunningham, P. (eds.) Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop on Case-Based Reasoning, EWCBRY98, Dublin, Ireland. pp. 196–207. Springer-Verlag, Berlin (1998)
4. Reinartz, T., Iglezakis, I., Roth-Berghofer, T.: Review and restore for case base maintenance. Computational Intelligence: Special Issue on Maintaining Case-Based Reasoning Systems 17(2), 214–234 (2001)
5. Richter, M.M.: The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal (1995)
6. Richter, M.M.: Introduction. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S. (eds.) Case-Based Reasoning Technology – From Foundations to Applications. LNAI 1400, Springer-Verlag, Berlin (1998)
7. Roth-Berghofer, T.: Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology, Dissertation. Ph.D. thesis, Universität Kaiserslautern (2003)
8. Roth-Berghofer, T.R.: Knowledge maintenance of case-based reasoning systems: the SIAM methodology, vol. 262. IOS Press (2003)
9. Smyth, B., McKenna, E.: Building compact competent case-bases. In: Althoff, K.D., Bergmann, R., Branting, L.K. (eds.) Case-Based Reasoning Research and Development: Proceedings of the Third International Conference on Case-Based Reasoning, ICCBR'99, Seeon Monastery, Germany. pp. 329–342. Springer-Verlag, Berlin (1999)