Case Base Maintenance: Terms and Directions

Fernando Orduña Cabrera and Miquel Sànchez-Marrè
Technical University of Catalonia (UPC)
Campus Nord-Building Omega Software Dept. (LSI)
Jordi Girona 1-3, E08034, Barcelona, Spain
{forduna, miquel}@lsi.upc.edu

Abstract. Since last years Case Base Reasoning (CBR) field has been growing, and Case Base Maintenance (CBM) is getting more important. Recent research has focused on case-base maintenance, addressing such issues as maintaining consistency, preserving competence, and controlling case-base grow. A set of dimensions for case-base maintenance proposed by Leake and Wilson, provides a framework for understanding and expanding CBM research. Taking this contribution into account, the aims of our work is to do a framework where the basics concepts of CBM are explained, and even more, as second objective we do a brief resume of some relevant contributions made by the scientific CBR community. Starting where Wilson and Leake research work ends.

Keywords. Case Base Reasoning, Case Base Maintenance, Policies.

1. Introduction

Janet Kolodner in [28] define Case-Based Reasoning (CBR) as: "Case-based reasoning can mean adapting old solutions to meet new demands, using old cases to explain new situations, using old cases to critique new solutions, or reasoning from precedents for interpret a new situation (much as lawyers do) or create an equitable solution to a new problem (much as labor mediators do)".

CBR systems solve new problems by retrieving and adapting the solutions to previously solved problems that have been stored in a case-base. The performance of a case-based reasoner can be measured according to Efficiency - Competence - Quality.

As CBR systems are deployed in real-world situations the issue of case maintenance becomes more and more critical. Uncontrolled case-base growth can cause serious performance problems as retrieval efficiency degrades and incorrect or inconsistent cases become increasingly difficult to detect.

Maintenance in CBR can mean a number of different things: out-of-date, redundant, or inconsistent cases may be deleted; groups of cases may be merged to eliminate redundancy and improve reasoning power; cases may be re-described to repair inconsistencies.

Case-Based Maintenance (CBM) has become an active CBR research area, producing results with important ramifications for both the theory and practice of CBR. Much significant work in this area focuses on developing methods for reducing the size of the case-base while maintaining case-base competence. The goal of achieving compact competent case-bases addresses important performance objectives for CBR systems. As an added benefit, compact case-bases decrease communications costs when case-bases are used as vehicles for knowledge sharing or are transferred in distributed CBR systems. However, case-base compactness is only a proxy for performance in a CBR system, rather than an end in itself.

Experience with the growing number of large-scale CBR systems has led to increasing recognition of the importance of case-base maintenance. Multiple researchers have addressed pieces of the case-base maintenance problem, considering such issues as maintaining consistency and controlling case-base growth.

The framework of CBM shown in [12,7] are useful to understand the state of the art in case-base maintenance. Those research works presents a first attempt at identifying the dimensions of case-base maintenance. Its shows that characterizations along such dimensions can suggest avenues for future case-base maintenance research and presents initial steps exploring one of those avenues: identifying patterns of problems that require generalized revisions and addressing them with lazy updating.

In [12,7] deals with a brief description of some research works done before 2001. In our research work, in the section two a description of the TERMS involved in the CBM field are seen. In section three a brief description of relevant research works in the field of CBM are shown.

2. Terms of Case-Based Maintenance

Case-Base Maintenance is defined by David Lake such as the process of refining a CBR system's case-base to improve the system's performance: ***"Case-base maintenance implements policies for revising the organization or contents (representation, domain content, accounting information, or implementation) of the case base in order to facilitate future reasoning for a particular set of performance objectives"***.

Maintenance in CBR can mean a number of different things: out-of-date, redundant, or inconsistent cases may be deleted; groups of cases may be merged to eliminate redundancy and improve reasoning power; cases may be re-described to repair inconsistencies.

Thus case-base maintenance may involve revising indexing information, links between cases, or other organizational structures and their implementations.

Maintaining case-base contents may affect a single case or multiple cases. It may revise

- The case representations used
- Either domain information in the case-base or accounting "information"
- How case representations are implemented
- The case-base at the implementation level, representation level, or the knowledge level

Performance objectives provide criteria for evaluating the internal behavior and task performance of a particular CBR system for a given initial case-base and sequence of problems solved. The performance objectives may be quantitative or qualitative. Performance objectives may change over time to reflect varying external circumstances, which may necessitate changing (maintaining) maintenance policies as well. Performance models which combine competence and efficiency can be used to guide the deletion of redundant cases from a case-base in order to optimize system performance.

Effective maintenance relies on an ability to model the complete performance characteristics (both *competence* and *efficiency*) of a system and case base.

Efficiency-Directed Maintenance *Utility problem:* The utility problem highlights the link between knowledge-base (case-base) size and the retrieval time needed to select an item of knowledge to use in a particular problem solving situation. Addition of more knowledge results in potentially severe efficiency degradation.

Utility metric is defined which takes into account the cost of maintaining the knowledge item (the *retrieval or match cost*) and the expected problem solving savings offered by the item (*average savings* multiplied by *application frequency*).

Competence Directed Maintenance Individual knowledge items only contribute to problem solving efficiency. An underlying first-principles problem solver is always used to encode basic problem solving competence. Cases contribute to both competence and efficiency.

Effective maintenance in case-based reasoning depends on the ability to measure and manage case competence as well as case efficiency.

Modelling Case Competence Competence means the range of problems that can be satisfactorily solved.

Coverage of a case: the set of target problems that a given case can successfully solve. ***Reachability*** of a target problem: the set of cases that can be used to solve a given target problem.

Cases with large coverage sets seem likely to be making large competence contributions. In contrast, cases that are members of large reachability sets seem likely to be less important, as many other cases exist which can solve similar problems. The ability to measure coverage and reachability is the key to understanding competence in CBR. Of course it should be clear that the coverage and reachability sets depend on the characteristics of particular retrieval and adaptation methods.

Definition 1 - Case Coverage A case-base

$$C = \{c_l \dots c_n\}, c \in C, Coverage(c) = \{c' \in C : Adaptable(c, c')\}$$

Definition 2 - Case Reachability A case-base

$$C = \{c_1 \dots c_n\}, c \in C, Reachable(c) = \{c' \in C : Adaptable(c', c)\}$$

Competence Categories: A ***pivotal case*** covers a region of the problem space which is otherwise uncovered.

At the other extreme is the ***auxiliary case*** and has no competence contribution to offer, since its coverage set is subsumed by the coverage set of another case.

Support Cases exist in groups and offer the same coverage as other cases in a group.

During future problem solving, as cases are learned and deleted from the case-base, the case categories must be updated by re-computing the coverage and reachability of affected cases to adjust the categories accordingly.

Describing CBM Policies The goal of a categorization scheme for case-base maintenance is threefold. First, by identifying classes of similar maintenance approaches, such a categorization scheme can shed light on the state of current practice in the field, increasing understanding of current CBM approaches. Second, mapping out the space of candidate approaches helps identify parts of the space that have not been addressed in previous work; these gaps in turn suggest research opportunities. Third, a categorization scheme for maintenance approaches is a first step towards cataloging the approaches that are most appropriate for particular performance goals.

Maintenance policies are described in terms of how they gather data relevant to maintenance, how they decide when to trigger maintenance, the types of maintenance operations available and how selected maintenance operations are executed.

Data collection gathers, synthesizes, and distills the data about the case base and about system processing. Gathers information about

- Individual cases might record the number of times a case has been successfully used or the number of times it has failed.
- The case base as a whole could involve, for example, monitoring the size of the case base.
- Processing might involve noting clusters in input problems or input problems that the system is unable to solve successfully.

Triggering takes this information as input, makes the decision whether maintenance is needed, and selects maintenance actions from a range of possible operation types. The results of data analysis serve as input for determining whether case-base maintenance is necessary. Both the *timing and integration* dimensions apply to this step as well.

Conditional triggering can be subdivided into three classes depending on the conditions that determine whether maintenance is triggered: *space-based* limited amount of case storage, *time-based*, retrieval time exceeding a threshold, or *result-based*, the system failing to solve a given problem or the wrong case being retrieved.

Execution describes how the selected revisions are actually applied to the case-base.

Type of data: None, Synchronic, or Diachronic:

- To decide when case base maintenance is needed.
- Is to do no collection at all.
- This type of policy is referred to as non-introspective.

More sophisticated reasoning is enabled by considering a snapshot of the current case base in part or as a whole. Policies that consider snapshot information are called synchronic.

The most informative approach is to collect data over time, over a sequence of snapshots, in order to identify trends in how case-base contents and usage are changing. Policies that consider changes in the case-base over time are called diachronic.

Timing: Periodic, Conditional, or Ad Hoc: A maintenance policy must specify when data collection is performed. *Periodic timing* happens at a set frequency with respect to the CBR cycle is termed *continuous*. Conditional data collection is performed in response to a well-defined but non-periodic condition. *Ad hoc* timing happens under ill-defined conditions determined externally to the CBR system.

Integration: On-line or Off-line: Data collection may operate on-line, during the course of an active reasoning episode, or off-line, during a pause in reasoning, such as waiting for user input or when idle between reasoning episodes.

Operation types: Different maintenance policies revise different types of information (the target type) at different levels (the revision level).

Target type: Revision operations can focus on four types of targets: Indexing structures, domain contents, accounting information.

Revision level: Revision operations can make revisions at three levels: The implementation level, the representation level, knowledge level.

Execution: Execution is characterized by the timing of maintenance operations and their integration with other system processing. Execution timing is described using data collection (periodic, conditional, or ad hoc). Execution integration is described as on-line or off-line depending on whether maintenance operations are performed during or between reasoning episodes.

Scope of Maintenance: Broad or Narrow: Operations that affect a single case or a small subset of the case-base have narrow scope, and operations that affect a large subset or the entirety of the case base have broad scope.

The standard learning of CBR (always adding each new case to the case base):

- Non-introspective
- Continuous (periodic) and on-line
- The scope of change is narrow
- Non-introspective
- Timing is ad hoc
- Integration is off-line
- The scope varies from narrow to broad.

Categorizing Policies for CBM: *Policies targeting domain content*: May be divided into policies aimed at adding and deleting cases and policies aimed at revising internal case content. *Standard case learning and manual maintenance*: always adding each new case to the case base. *Additional policies aimed at case retention*: based on coverage and reachability, and integrating offline or online, beneficial or detrimental. *Policies aimed at interval case content*: Are aimed at internal case content. *Policies targeting indices*: A number of classification systems using IBL and related techniques IBL_n include policies for eliminating noisy and redundant instances from a set of training examples (cases). *Policies targeting maintenance policies* include the capability for *meta-maintenance* maintenance of the maintenance strategies themselves.

3. Contributions to CBM Field

The research works [16,17,29], has the goal of to analyze Case Base Management strategies in the context of a multimodal architecture, combining CBR and Model Base Reasoning (MBR), and present the follows policies in the research:

Replace Policy *(is a competence-based strategy, Replace a set of stored cases with the current one if the letter exhibits an estimated competence comparable with the estimated competence of the considered set of stored cases)*: based on competence model. Replace a set of stored cases with a case to be added to the CB.

Is the core of the adaptation-guided retrieval algorithm and is considered as the main principle for the definition of a competence-based strategy of CB strategy of CB revision.

Learning by failure with forgetting LFF *(aims at forgetting cases whose usage does not fulfill specific utility conditions. Present an alternative deleting strategy based on notion of usefulness, a case is useless when it has either never been retrieved or* *never been adapted with success. Does not take competence into direct consideration. Requires the definition of a notion of time (discrete))*: based on incremental learning of cases interleaved with offline processes of forgetting.

The primary mechanism for leaning by experience is the ability for learn the solution to a problem trough MBR when CBR has been unable to solve it. In both policies over-growth will not occur and significant speed up competence and quality are in significant levels.

In [10,20,19],propose an extended six-step CBR cycle. The two additional steps as part of the maintenance phase of CBR process. Define several quality measures that describe characteristics of CB such as correctness, consistency, uniqueness, minimality and incoherence. Describe methods for modifications of CB in the restore step and their relation to the review steps:

- Review step: cover tasks to judge and monitor the current state of CBR system and its knowledge containers. Define several quality measures based on case and CB properties as well as a set of example monitor operators to control the quality of the CB.
- Restore step: invokes mechanisms to change the system and its knowledge in order to return to a usable state in situations where CBR system performance does not met desired requirements anymore. Define different modify operators and discussed their relation to the review step.

Smyth and Keane in [2], examines a deletion strategy in the context of CBR systems. Investigates a number of traditional deletion policies and their application to CBR. Introduce 2 new policies designed specially with CBR in mind the policies, recognize the possibility of competence degradation trough case deletion and safe-guard against it by using an explicit model of case competence to guide deletion.

Introduce ***Coverage and Reachability***: **Coverage**: is the set of targets problems that it can be used to solve it. **Reachability**: is the set of cases that can be used to provide a solution for the target.

Introduce *Pivotal cases, Auxiliary cases, Spanning cases, Support cases*, and its definitions.

Introduce the algorithms: ***Learning Update***: for modeling case competence propose a heuristic for efficiently updating the case categories. ***FootPrint Deletion Policy***: provides the same competence as the entire CB but with fewer cases. Deletion Update> all procedures are estimates; they make assumption that the space of target problems is accurately approximated by CB. ***FUD***: footprint-utility deletion combining footprint deletion and utility deletion.

Combining competence and performance, can be used to select between a number of alternative auxiliary cases or between a number of support cases belonging to the same support group or between a numbers of pivotal cases.

The solution proposed uses a model of cases competence to guide the learning and deletion of cases.

Smyth and McKenna in [6], focused on the competence properties of cases and developing explanatory and predictive models of case competence that can provide a sound foundation for feature maintenance solution. Provides a survey of the work on the development of explicit algorithmic models of competence for CBR systems. Explain how explicit competence models can be used to construct a competence map of an evolving CB and how such a map can be used to identify both regions that are competence rich and those that are competence poor.

Competence

The Foundations of Competence: the contribution of individual cases can be characterized by two sets: *Coverage set* of a case is the set of all target problems that this case can be used to solve. *Reachability set* of target problem is the set of all cases that can be used to solve it. *Competence Groups*: Coverage and reachability sets provide a measure of local competence only. Each group makes a unique contribution to competence. *Footprint and Relative Coverage*: Estimates the competence contribution of an individual case *c* as a function of the size of the case's coverage set. Evaluated the Footprint-Creation Algorithms:

- CNN-FP footprint creation algorithm
- RC-FP footprint creation algorithm
- RFC-FP footprint creation algorithm
- COV-FP footprint creation algorithm

Barry Smyth in [3], propose a novel competence-based maintenance policy for CBR systems. His paper is basically similar to the paper [2]. The paper is focusing on explain in-depth the cases introduced on [2], and only present the FootPrint Deletion Algorithm but with a little change on it. On this algorithm the Spanning cases are not taking in count.

The research works [13,11] present case addition maintenance policy that is guaranteed to return a concise CB with good coverage quality and case addition algorithm. They results highlight benefit reduction as a key factor in influencing the coverage of CB coverage when add a case to the CB. The addition-based policy can place a lower bound on the coverage of the resulting case. The aim of the paper is to find a near-optimal CB of size K efficiently. Present a different CBM policy that is based on case addition rather than deletion. CBM is divided into two broad categories: maintaining the CB indixes and maintaining the CB contents. The case addition-based policy (CABP) a case is good if its neighborhood is large. Introduce terms such as frequency function, benefit, optimal CB and near-optimal CB. The maintenance policy should select CB on the size of the neighborhood that each case has an a CB, therefore a case should be selected first into a new CB, this step is critical for the quality of a CB. CABP functionality is Case-Addition Algorithm (CAA) and is a greedy algorithm. CAA: CAA produce X_1 the coverage of X_1 is no less than 63% of the coverage of an optimal CB. Adaptation cost is based on this cost define the neighborhood.

The purpose of the paper [14], is to present a novel case-base maintenance and retrieval system aimed at improving the accuracy and performance of a CBR system when the number of cases gets large. The idea is to create multiple, small case bases that are located on different sites. Each small case base contains cases that are closely related to each other. The approach is to allow the cases to be added and deleted at each small case base without affecting the whole. Propose a case-base maintenance method that avoids building sophisticated structures around a case base or perform complex operations on a case base. The clustering method can also be based on values of attributes that may be attached to the cases.

To support retrieval present a method that is based on a decision forest built with the attributes that are obtained through an innovative modification of the ID3 algorithm. The approach is based on two related ideas. The *first one is clustering*, whereby a large case base is decomposed into groups of closely related cases. Second idea is to allow a user to *retrieve* the distributed case bases by incrementally selecting the attributes that are information-rich and can cover the entire distributed case base structure. Choose a density-based clustering method as the basis because it is relatively efficient to execute and does not require the user to pre-specify the number of clusters. Developed a method for finding a near optimal *Eps* value through a local search process. They call this new algorithm ***CBSCAN***. *CBSCAN* is based on the observation that the minimum radius value *Eps* is critical in determining the quality of a partition. They use the new *Condorcet criteria (NCC)*, is based on the idea that for a partition to be good cluster is a good partition has small intra-cluster distances and large inter-cluster distances.

In [15] presents CBM policies for case index revision and case retention in the context of CB planners performing CB adaptation by derivational replay. The policies are based on the outcome and the benefits in the retrieval cases. The two policies are:

Case-Index Refinement Policy: with propose of to tune the feature weight according to relative importance of a feature in particular case. Presents the terminology case, problem, features (associated with the weight) and goals. Steps of the Step Index Revision:

1. The outcome of the retrieval is stated for C
2. The set of $feat_c$ of all features in C that did not occur in the new problem is determined.
3. The weight of each feature in $feat_c$ is revised.

The features are update by the times of on C is extensible using counters.

Case Retention Policy: Based on the contribution of the retrieved cases to the overall adaptation effort. The effort is measured in terms of size of the search space that was retrieved to solve the problem.

Follows two values: Size Case and Size Plan.

A retrieval is beneficial *if Size Case/(Size Case + Size Plan) >= thr* Conclude: Retaining cases bases on the benefit of the retrieval is a more adequate policy than retaining nonextensible cases, and the improvement there is a mainly due to standard CBR guidance.

The paper [7,12], presents a first attempt at identifying the dimensions of case-base maintenance. Presents a first step in the direction of a general CBM framework. Define case-base maintenance as the process of refining a CBR system's case-base to improve the system's performance:

Case-base maintenance implements policies for revising the organization or contents (representation, domain content, accounting information, or implementation) of the case-base in order to facilitate future reasoning for a particular set of performance objectives.

Thus case-base maintenance may involve revising indexing information, links between cases, or other organizational structures and their implementations.

Policies The goal of a categorization scheme for case-base maintenance is threefold. First, by identifying classes of similar maintenance approaches, such a categorization scheme can shed light on the state of current practice in the field, increasing understanding of current CBM approaches. Second, mapping out the space of candidate approaches helps identify parts of the space that have not been addressed in previous work; these gaps in turn suggest research opportunities. Third, a categorization scheme for maintenance approaches is a first step towards cataloging the approaches that are most appropriate for particular performance goals.

Data collection about individual cases might record the number of times a case has been successfully used or the number of times it has failed.

Triggering The results of data analysis serve as input for determining whether case-base maintenance is necessary.

Operation Types Different maintenance policies revise different types of information (the target type) at different levels (the revision level).

Execution is characterized by the timing of maintenance operations and their integration with other system processing.

This paper presents an initial framework for characterizing case-base maintenance policies. It presents basic dimensions for CBM policies in terms of three sub-processes; data collection, triggering, and execution and characterizes key design choices in terms of those dimensions.

The paper [9], has presented an argument for integrating performance considerations more directly in to case addition and deletion procedures, in order to allow finer-grained optimization of case-base contents. The paper shows that the relationship between *competence, compactness* and *adaptation performance* is more subtle than a simple tradeoff in some circumstances, adaptation performance can be increased without sacrificing competence or compactness motivating the search for ways to refine case addition and deletion procedures to improve performance results. Just as the direct connection of retrieval criteria to adaptation abilities led to important progress, the direct connection of case-base construction to performance criteria promises important advances for case-base maintenance research.

Effective maintenance requires remembering why cases are being remembered (or forgotten) to serve the overall performance goals of the CBR system for a given task and optimizing maintenance decisions accordingly.

This paper examines the benefits of using fine-grained performance metrics to directly guide case addition and deletion, and presents initial experiments on their practicality.

Their results show that performance-based deletion strategies are especially promising for non-uniform problem distributions, which have received little attention in previous analysis of case-based maintenance, but which are often important in real-world contexts. Case-base maintenance is fundamentally driven by performance concerns.

Describes a strategy for performance-based case selection, inspired by Smyth and McKenna's RC-CNN algorithm. That algorithm compacts case-bases using a compressed nearest-neighbor (CNN) algorithm whose inputs are ordered by a relative coverage (RC) metric, to give priority to cases expected to make the largest competence contributions. Developed a relative performance (RP) metric aimed at assessing the contribution of a case to the adaptation performance of the system.

Explored a number of metrics, including a "performance benefit" (PB) metric estimating the actual numerical savings that the addition of each case provides.

In [21] consider the learning of the retrieval knowledge (organization) as well as the prototypes and the cases as case-based maintenance. They address this problem based on cases that have a structural case representation. Propose a similarity measure for an attributed structural representation and an algorithm that incrementally learns the organizational structure of a case base. This organization schema is based on a hierarchy and can be updated incrementally as soon as new cases are available. Describe two approaches for organizing the case base based on approximate graph subsumption. First approach (*uses a fixed threshold for the similarity values*) is based on a divide-and-conquer strategy whereas the second (*uses for the grouping of the cases an evaluation function. Use a strategy more flexible to fit the hierarchy dynamically to the cases BUT NOT allow to incorporate new cases into the hierarchy and open new nodes by splitting the leaf nodes into two child nodes*) one is based on a split-and-merge strategy which better allows to fit the hierarchy to the actual structure of the application. Use an image-related application to show how it can be used for matching.

The derived hierarchy consists of nodes and edges. Each node in this hierarchy contains a set of cases that do not exceed a specified similarity value. The edges show the similarity relation between the nodes. Based on intra-class similarity it is decided whether a case is to be removed from or to be stored in a cluster.

In the paper do not consider case deletion, but focus instead on the addition of new cases to the case base. Only propose a similarity measure for structural representations which can handle feature weights, but we will not consider how these feature weights can be learned. Instead, they will assume that the user can specify this feature weight a priori. The structural representation of a case can be described as a graph.

Graph subsumption, on the basis of the part isomorphism, we can introduce a partial order over the set of graphs. And consider an algorithm for determining the part isomorphism of two graphs. That's match the graph.

Subsumption is a sub set relation between the extensions of concepts and allow observing the conceptual knowledge of the domain of the interpretation.

A *graph* is defined by the index structure of the CB. *Transitivity* is used like a part of isomorphism for the reductions of nodes that have to be computed. *Retrieval* Is done by classifying the current case through the index hierarchy until node represented by a prototype. *Utility* a heuristic is used to evaluate the partitions. *Prototype* learning evaluates de variance of the graphs in one case class and another is to calculate the median of the case in a case class. *Algorithm* Adapt the notion of Fisher for concept learning.

Niloofar Arshadi in [30,31], propose a maintenance technique that integrates an ensemble of CBR classifiers with spectral clustering and logistic regression to improve the classification accuracy of CBR classifiers on (ultra) high-dimensional biological data sets. Maintenance method improves the classification accuracy of TA3.

Main challenge is to interpret the molecular biology data to find similar samples to ventually use them in case-based medicine, and to identify those genes whose expression patterns have meaningful relationships to their classification labels.

CBR maintenance approach has three main components: *ensemble of CBR systems, clustering, and feature selection.* They use an ensemble of CBR systems, called *mixture of experts* (MOE) to predict the classification label of a given (input) case.

Apply *spectral clustering* to cluster the data set into *k* groups, and the logistic regression model is used to select a subset of features in each cluster. Used as a filter feature selection for the TA3 classifier.

The goal of *MOE4CBR Method* is to improve the prediction accuracy of CBR classifiers, and at the same time reduce the size of the case-base knowledge container. According to Smyth's categorization. Are both competences directed and efficiency-directed. *Competence-directed* maintain the case-base to provide the same (or better) quality solution to a broader range of problems. Efficiency-directed consider the processing constraints, and modify knowledge containers to improve efficiency of storage or scalability of retrieval.

Feature Selection Applied there feature selection technique to TA3 and those were Fisher criterion, t-test, and logistic regression model. Logistic regression outperforms the others. Used as filter the data ser were normalized.

Each expert classifies samples separately, and individual responses are combined by the gating network to provide a final classification label.

MOE4CBR maintenance method has two main steps, is a generic technique for improving the prediction accuracy of CBR classifiers:

1. CB of each expert if formed by clustering the data set into k groups.
2. The each CB is maintained locally using future selection.

Each of the *k* obtained sets will be considered as a case-base for our *k* CBR experts, then they combine the responses and each expert applies the TA3 model to decide on the class label, and the gating network uses TA3 to assign weights to each classifier, i.e., to determine which class the input case most likely belongs to.

In [22] present two approaches based on deletion policies to the maintenance of case memories. The foundations of both approaches are the Rough Sets Theory, The main purpose of these methods is to maintain the competence of the system and reduce, as much as possible, the size of the case memory. The aim of this paper is twofold: (1) to remove noisy cases and (2) to achieve a good generalization accuracy. Defining a competence model based on Rough sets and presenting new hybrid approaches to improve the weak points. Present two hybrid approaches: ***Accuracy-Classification Case Memory (ACCM) and Negative Accuracy-Classification Case Memory (NACCM)***.

Two approximations are generated: The lower approximation $\underline{R}X$ elements can certainly be classified. Upper approximation $\bar{R}X$ elements which can possibly be classified.

The difference between them is to facilitate the usage of coverage when selecting cases that are deleted from the original case memory. The aim of reduction techniques is to take advantages of the benefits of each coverage measure. ACCM the main idea is to benefit from the advantage of both measures separately. NACCM is based on ACCM doing a complementary process the main is to select all cases that are near to the outliers and maintain those cases that are completely internal and do not have any cases whose competence are contained. Test UCI repository and their own repository the test are not reduced to much. The aim is to maintain the minimal set of cases in the case memory.

Salamo in [23] continue the research of [22], where presents a model that allows to update itself dynamically taking information from the learning process. Different policies has been applied to test the model. Introduces a dynamic case base maintenance (DCBM) model that updates the knowledge (case base in CBR) based on the learning problem solving process. The knowledge update is based on Reinforcement Learning. Use a *Monte-Carlo* method because is the only one that use experience of the environment to learn the value functions. The Bellman equation is applied: In order to find the optimal value functions. In the model the reinforcement function is the revise phase of the CBR cycle. The value function is used and modified by the RL algorithm to learn the optimal case base. Action is to delete or maintenance a case from CB. Enviroment is the CBR cycle. Coverage defined as the initial sum of features reward using Rough Set measures. DCBM model the retrieval phase selects KNN in order to accelerate the maintenance process.

The CORE of the RL process in the CBM policy function such as: *Policy RLOLevel* Reinforcement learning obvilon policy by level of coverage, gives an algorithm for these policy, based on ACCM that policy will be very aggressive with the CB because it maintain the minimum description of the CB. They believe my not work property in a dynamic environment.

Policy ROLCE Reinforcement learning obvilon by coverage and error the coverage is the relevance of a case, show the simplest way to decide the actions, is based on coverage lost.

In [25] three algorithms are proposed for maintaining a case base. The diversity of cases (relative to each other) plays an important role in these new algorithms. Combine similarity and diversity of cases together for determining a new case base. Reduce the size of a case base without comprising its competence, especially for the algorithm based on evolution strategies (**ESs**). The diversity (denoted as **RelDiversity**) of cases is viewed as the average dissimilarity between all pairs of these cases.

Maintenance algorithms: **Random-based selection**: This is a simpler algorithm (Random), which is used for selecting *k* cases randomly from *C*. The randomness indicates the relative diversity of selected cases is indirectly considered.

RelDiversily-Based Selection: *k* cases are selected by incrementally building a retrieval set, R. During each step, the remaining cases are ordered according to their RelDiversity and the case with highest RelDiversity is added to *R*. Note that the key to this algorithm is the *RelDiversity* metric.

Evolution Strategies-Based Selection: a type of evolutionary algorithm. This selection algorithm is based on (1+1) strategy, and used for selecting *k* cases from n initial retrieved cases. Consider *k* parameters, each of which represents one case. Moreover, the value *x* for each parameter is determined by the sequence number of each selected case and the number of retrieved cases *n*.

The paper [26], mainly discusses how to maintain case bases in CBR system by adopting outlier data mining and case sieving techniques. Proposed algorithm can maintain case bases satisfactorily and stably. Paper puts forward a method to maintain case base by using outlier mining and case sieving techniques. The algorithm first extracts the outliers, then eliminates the noise from them according to the domain knowledge, and finally stores the non-noise data as a particular category.

Are five important approaches that can be used in outlier mining:

- Statistics-based outlier mining technique
- Deviation-based outlier mining technique
- Rule-based outlier mining technique
- Distance-based outlier mining technique
- Clustering-based outlier mining technique

The algorithm to detect erroneous cases from outliers is presented below:

1. Determine the property set that need to be detected and establish the similarity function;
2. Find out the case subset that has the least similarity with others.
3. Examine these cases within the given property set to see whether they are erroneous or not.

Presents the algorithm to detect erroneous cases from outliers. They choose distance-based outlier mining method to discover outliers. Objects that don't have many neighborhoods are considered outliers. Distance measurement algorithm with weight is an improvement on KNN which assigns a certain weight to each property in the data set, the weight indicates the importance of a property in the future space. The maintenance algorithm can be divided in three phases: *Extracting* outliers from existing case bases; *analyzing* and eliminating erroneous outlier cases; *sieving* cases from non-outliers.

Found that our method is stable than other approaches, because the accuracy is dropping slowly, while others dropping sharply. And more, our algorithm has better accuracy than other methods.

4. Classification Table

For a clasification see the next table where:
a = propose add tow steps, Review and Restore; b = Clustering and density-based; c = Present a general framework of CBM; d = Spectral Ultra high-dimensional data; e = Reinforcement learning; f = Combine similarity and Diversity.

Reference	Learning	Deleting	Addition	Indexes	Competence	Retrieval	Data Mining	Algorithms
[16,17,29]	Incremental	x			x	x		retrieval
[10,20,19]	a							
[2]		x			x			x
[6]		x						x
[3]					x			
[13,11]	x		x	x				
[14]						x	b	
[15]	x			x		x		
[7,12]	c							
[9]		x	x					
[21]	x					x		
[30,31]							d	
[22]		x					outliers	
[22,23]	e							
[25]								
[26]							outliers	

5. Conclusions

Some years ago in Leake and Wilson were proposed a research work, with the aim of explain how CBM could improve the cycle of CBR. This research work presents a definition for CBM. Using the CBM definition and their contributions, many research works have been working in new policies to improve the CBR cycle. The report done is focused in to do a framework where the basics concepts of CBM are explained and a brief resume of some relevant contributions. With this report and the research works of Leake, we find how the contributions in the CBM field have been growing, and how helps to improve in CBR policies.

References

[1] Barry Smyth and Elizabeth McKenna, "Footprint-Based Retrieval". In Case-Based Reasoning Research and Development. Lecture Notes in Artificial Intelligence. Edited by K.D. Althoff, R. Bergmann and L. Branting. Springer-Verlag Berling. pp. 343-357, 1999.

[2] Barry Smyth and Mark T. Keane, "Remembering To Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems". Proceedings of the 14'h International Joint Conference on Artificial Intelligence, pp. 377-382. Canada.

[3] Barry Smyth, "Case-Base Maintenance". Proceedings of the 11th Intemational Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 1998.

[4] B. Smyth and P. McClave, "Similarity vs. diversity". Proceedings of the 4th International Conference on Case-Based Reasoning, pp. 347-361, 2001.

[5] Smyth, B., and Cunningham, P. "The utility problem analysed: A case-based reasoning perspective". In Proceedings of the Third European Workshop on Case-Based Reasoning, 392–399. Berlin: Springer Verlag. 162. 1996.

[6] Barry Smyth and Elizabeth McKenna, "Competence Models and the Maintenance Problem". Computational Intelligence, vol. 17(2), pp.235–249, 2001.

[7] Leake, D., and Wilson, D. "Categorizing Case-base maintenance: Dimensions and directions". In Cunningham, P.; Smyth, B.; and Keane, M., eds., Proceedings of the Fourth European Workshop on Case-Based Reasoning, 196–207. Berlin: Springer Verlag. 1998.

[8] Leake, D., and Wilson, D. "When experience is wrong: Examining CBR for changing tasks and environments". In Proceedings of the Third International Conference on Case-Based Reasoning, 218–232. Berlin: Springer Verlag. 162, 166. 1999.

[9] D.B. Leake, and D.C. Wilson, "Remembering why to remember: performance-guided case-base maintenance". Proceedings of the Third Intemational Conference on Case-Based Reasoning, pp. 161–172, 2000.

[10] T. Reinartz and I. Iglezakis, "Review and restore for case-base maintenance". Computational Intelligence, vol. 17(2), pp.214–234, 2001.

[11] Jun Zhu and Qiang Yang, "Remembering to add: Competence-preserving case-addition policies for case base maintenance". In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann. 161, 163, 165. IJCAI 1999.

[12] D.C. Wilson, and D.B. Leake, "Maintaing case-based reasoners: dimensions and directions". Computational Intelligence;vol.l7 (2), pp. 196–213,2001.

[13] Qiang Yang and Jing Zhu, "A case-addition policy for case-base maintenance". Computational Intelligence, vol. 17(2), pp. 250–262, 2001.

[14] Qiang Yang and Jing Zhu, "Keep It Simple: a Case-Base Maintenance Policy Based on Clustering and Information Theory". Canadian AI2000, LNAI 1822, pp. 102–114, 2000.

[15] Héctor Muõz Avila, "Case Base By Integrating Case-Index Revision and Case-Retention Policies en a Derivational Replay Framework". Computational Intelligence, vol. 17(2), pp. 280–294, 2001.
[16] Luigi Portinale, Pietro Torrasso, "Case-Base Maintenance in a Multi-modal Reasoning System". Computational Intelligence, vol. 17(2), pp. 263–279, 2001.
[17] "Speed-up, Quality and Competence in Multi-modal Case-Based Reasoning", ICCBR'99, LNAI 1650. pp. 303-317.
[18] Thomas Reinartz, Ioannis Iglezakis and Thomas Roth-Berghofer, "Review and Restore for Case-Base Maintenance". Computational Intelligence, vol. 17(2), pp. 214–234, 2001.
[19] Ioannis Iglezakis, Thomas Reinartz and Thomas Roth-Berghofer, "Maintenance Memories: Beyond Concepts and Teachniques for Case Base Maintenance". ECCBR 2004, LNAI 3155, pp. 227–241. 2004.
[20] Thomas Reinartz, Ioannis Iglezakis and Thomas Roth-Berghofer, "On Queality Measures for Case Base Maintenance". EWCBR 2000, LNAI 1898, pp. 247–260. 2000.
[21] Petra Perner, "Case-base maintenance by conceptual clustering of graphs". Engineering Applications of Artificial Intelligence, vol. 19 (2006), pp. 381–393.
[22] Maria Salomó and Elisabet Golobardes, "Hybrid Deletion Policies for Case Base Maintenance". American Association for Artificial Intelligence (aaai) 2003.
[23] Maria Salomó and Elisabet Golobardes, "Dynamic Case Base Maintenance for Case-Based Reasoning System". IBERAMIA 2004, LNAI 3315, pp. 93–103, 2004.
[24] Maria Salomó and Elisabet Golobardes, "Deleting and Building Sort Out Techniques for Case Base Maintenance". ECCBR 2002, LNAI 2416, pp. 365–379, 2002.
[25] Lan-Zhen Yang, Ming-Hu Ha, Xi-Zhao Wang, "Diversity-Based Case Base Maintenance". Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2–5, November 2003.
[26] Zhi-Wei Ni, Yu Liu, Feng-Gang Li, Shan-Lin Yang, "Case Base Maintenance Based on Outlier Data Mining". Proceedings of the Fourth Intenational Conference on Machine Learning and Cybernetics, Guangzhou, 18–21, August 2005.
[27] Rong Pan, Quing Yang, Sinno Jialin Pan, "Mining competent case bases for case-based reasoning". Artificial Intelligence vol. 171 (2007), pp. 1039–1068. 2007.
[28] Janet Kolodner, "Case Based Reasoning". Book Morgan Kaufmann Publishers, Inc. 1993.
[29] Maarten Van Someren, Jerzy Surma and Pietro Torasso, "A Utility-Based Approach to Learning in a Mixed Case-Based and Model-Based Reasoning Architecture". Lecture Notes in Computer Science. Case-Based Reasoning Research and Development. pp. 477–488. 1997.
[30] Niloofar Arshadi and Igor Jurisica, Maintaining Case-Based Reasoning Systems: A Machine Learning Approach, ECCBR 2004, pp. 17 - 31, 2004.
[31] Niloofar Arshadi and Igor Jurisica, Data Mining for Case-Based Reasoning in High-Dimensional Biological Domains, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Volume. 17, Number 8, pp. 1127-1137, 2005.