

AUTO-INCREMENT OF EXPERTISE FOR FAILURE DIAGNOSTIC

Karim Haouchine*, Brigitte Chebel-Morello* and Nouredine Zerhouni*

* *Automatic Control and Micro-Mechatronic Systems Department, 24, Rue Alain Savary, 25000 Besançon, France
(Tel: +33 (0)3 81 40 28 01; e-mail: {karim.haouchine, bmorelle, zerhouni}@ens2m.fr).*

Abstract: We have developed a diagnostic help system dedicated to the maintenance of a supervised industrial system for pallets Transfer (SISTRE). This diagnostic help system is based on a Case-Based Reasoning approach (CBR). The expertise considered in this help system and formalized in the case form in a case-base must be updated, while taking account of its quality. In this objective we propose a method allowing on one hand to structure the case-base and on the other hand to auto-increment it. An experimental study is undertaken through references benchmarks as well as an application on SISTRE.
Copyright © 2008 IFAC

Keywords: Diagnostic, Case-Based Reasoning, Case-Based Reasoning Systems, Case-Based Maintenance, auto-increment, competence, performance.

1. INTRODUCTION

SISTRE is a Supervised Industrial System of pallets TRAnsfer for launching manufacturing ranges handled by robots. It represents a flexible production system which needs to be maintained. Therefore, a diagnostic and repair help system has been set up. It aims to identify the location and failing component and to give the appropriate repair action to the encountered problem. To carry out the diagnosis on SISTRE, a knowledge-oriented method has been deployed, the Case-Based Reasoning (CBR), a method largely used in the technical diagnosis. CBR is an approach to problem solving and learning, by reusing the solutions to similar problems stored as cases in memory called case-base (Aamodt and Plaza, 1994). In our study, we work on expert data which one will model in the form of case feeding the case-base. The case is composed of two parts: the description of the case and its solution. The case structure is adapted to the diagnosis:

Problem → Symptoms (description of a diagnostic situation).

Solution → Origins + Actions (maintenance strategy).

Thus a case describes a particular situation and contains several attributes defined by their values, suitable for the situation.

There are many systems based on CBR dedicated to the diagnostic that operate successfully. An early diagnostic system named "Creek" is dedicated to car failures presented in (Aamodt, 04). Other systems are interested in troubleshooting of aircraft engines as "Cassiopee" that is interested for Boeing 737 in CFM International (Bergmann, et al., 2003). A system "Chekmate" dedicated to industrial printers by Domino UK Ltd. is presented in (Grant, et al., 1996). Cheetham and Graf (1997) have set up a "FormTool"

system from General Electric Plastics sites, to define and produce the colors of plastic desired by a customer. A system of gas turbines failure diagnostic was developed in General Electric Energy at Atlanta (Devany and Cheetham, 2005).

CBR handles a cycle which is composed of different steps, and in this paper we are especially interested in three phases: The retrieval phase that identifies similar cases with the new problem to be solved, the maintenance phase of the case-base and the learning phase that evolves the case-base. This last step allows the system to solve new problems that may arise in the future.

Indeed, one of specificities of CBR, contrary to the conventional methods of artificial intelligence, is its ability to reason on a minimum number of cases, which will grow rich progressively with time thanks to the training step in the existing case base. So that this development does not deteriorate the quality of the system, a step of maintenance must be done.

Several authors specify that the maintenance of the CBR systems can be reduced to the case-based maintenance (Iglezakis and Roth-Berghofer, 2000). The knowledge of CBR systems relates directly to cases, affected by changes in knowledge sources. Thus, consulting the case-base may be the most appropriate approach to overtake maintenance operations (Leake and Wilson, 1998).

Strategies for maintenance and quality criteria of the case-base will be approached in paragraph 2. Through this study, a proposal for a case-base structuring method will be our first contribution. Then, after having maintained the case-base, it will be incremented by new cases. However this integration must be made under definite conditions in order to ensure the quality of the system. Consequently, several questions are raised. Which is the case to be retained among those which

were solved? How to index this case? How to introduce it by respecting the structure of the expert-base (delicate updated of the expertise)? Which is its contribution to improving the quality of expert-base?

As a result, an auto-increment algorithm will be proposed in paragraph 3. This algorithm will allow, from a partial expertise, the installation of a mechanism for updating expertise without damaging the organization and quality of the case-base. In the same paragraph, an evaluation with tests on references benchmarks has been completed and will be illustrated. Finally, the feasibility and the implementation results of the case-base structure and its auto-increment applied on the SISTRE will be addressed in paragraph 4.

2. CASE-BASED MAINTENANCE

Case-based 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 (Leake and Wilson, 1998). First of all, the state of the art on the principal work made in case-base maintenance will be approached. Thereafter, a case-base structuring method is proposed in the second part.

2.1 State of the art

The CBM approach can be divided in two policies, one concerning optimization and the other a case-based partitioning. The objective of these approaches is to reduce the case retrieval time. The optimization policy consists of deleting less relevant cases by following two strategies: addition and the deletion of cases. Whereas, the partitioning policy consists of dividing the case-base into several search spaces. This enables to select, in an increasing manner, the attributes which are rich in information and which can cover the structure of the case-base (Yang and Zhu, 2001).

Several criteria of case-base cases were proposed in order to carry out an evaluation concerning case-base. The important criteria that contribute to the evaluation of a case-base are: competence and performance.

- *Competence* is the range of target problems that can be successfully solved.
- *Performance* is the answer time that is necessary to compute a solution for case targets. This measure is bound directly to adaptation and result costs.

Two important competence properties are the coverage set and the reachability set. *Coverage* of a case is the set of target problems that it can be used to solve. *Reachability* of a target problem is the set of cases that can be used to provide a solution for the target.

Performance depends critically on the *accuracy* and the *storage* (number of cases stored in the case-base). Many CBR systems use retrieval methods whose efficiency is related to the case-base size, and under these conditions the addition of

redundant cases serves only to degrade efficiency by increasing retrieval time.

There are two strategies in the case-based optimization policy: the case addition and suppression strategies. The case addition strategy consists in building a reduced case-base by the successive addition starting from an initially empty case-base, according to a criterion to be maximized.

Smyth and McKenna, which present a method that uses an explicit case competence model based on notions of coverage and reachability. Their "relative coverage" (RC) metric, provides a precise measurement of competence contributions for individual cases. The RC metric, associated with the condensed nearest-neighbour (CNN) algorithm, permits to successively retain only those cases which are not solved by a case that has already been retained, in order to obtain a new reduced case-base (Smyth and McKenna, 1999).

Leake and Wilson developed a relative performance (RP) metric aimed at assessing the contribution of a case to the adaptation performance of the system (Leake and Wilson, 2000). To attain the benefit of adding the case to the case-base, they first assume that the similarity metric will accurately select the most adaptable case for any problem.

However, concerning the case deletion strategy, from a given case-base, this strategy values cases according to the criteria in order to be able to suppress and bring the case-base to a specific number of cases. There are several strategies scanning the entire case-base, the most important are:

- *Deletion based on case-base size and density* is a method proposed by Smyth and Keane that studies the case-base size, the density and the distribution of cases in a case-base. It tries to keep the homogeneity of the cases density (Smyth and McKenna, 1998).

- *Iterative Case Filtering Algorithm (ICF)* that iteratively removes a case whose absence produces better results as compared to retaining it. It repeatedly uses a deletion rule that removes cases whose reachability size is greater than that of the coverage until the conditions of the rule are not satisfied (Brighton and Mellish, 2002).

The majority of these methods do not give satisfying results concerning the optimization of the case-base size according to the studied criterion. Moreover, there are some methods which are difficult to implement, and those which are easy to implement don't give a realistic results.

A strategy based on cases categorization and the competence model, proposed by Smyth and Keane (Smyth and McKenna, 2002) is presented. The authors consider that the case-base itself is a sample of the underlying distribution of target problems. The key concepts in categorizing cases are coverage and reachability. Four categories of cases are considered (Smyth and Keane, 1995):

- *Pivotal Cases*: a case is pivotal if it is reachable by no other case but itself. Its deletion directly reduces the competence of a system.

- *Spanning Cases*: Spanning cases do not directly affect competence. They are so named because their coverage spaces link (span) regions of the problem space that are independently covered by other cases
- *Support Cases*: Support cases are a special class of spanning cases and again they do not affect competence directly. They exist in groups, each support providing similar coverage as the others in a group.
- *Auxiliary Cases*: A case is an auxiliary case if the coverage it provides is subsumed by the coverage of one of its reachable cases. The deletion of auxiliary cases only reduces the efficiency of the system.

No competence metric to our knowledge has been studied at this level contrary to methods from case addition strategy. Consequently, we propose a method based on a case-base optimization algorithm associated with a competence measure.

2.2 Proposed method

We propose a case deletion strategy method based on the case categorization developed in (Smyth and Keane, 1995), and adapted in this study to perform the selection of the representative cases. To categorize the case, an algorithm associated to a Competence Measure (CM) is presented in (Haouchine, et al., 2008). This CM measure is inspired by the relative coverage measure RC in (Smyth and McKenna, 1999) used in the case-addition strategy. The CM incorporating two criteria of coverage and reachability, gives an individual contribution to the case competence in relation to the size of the latter's coverage set, while attributing to each coverage and reachability case a value that we shall name *coverage value* " Vc " (cardinal of target cases set in relation to source case) and *reachability value* " Vr " (cardinal of source cases set in relation to target case).

$$CompetenceMetric(c) = \frac{Vc(c)}{Vr(c)} \quad (1)$$

In order to have a case-base with good competence, its coverage ratio must be high and its reachability rate must be low. Consequently, the CM is used to guide the deletion of cases in the case-base by favouring the cases with a high CM value and deleting those with smaller CM value. Due to this fact, our method consists of reducing the case-base size while maintaining a maximal competence. The case categories will be determined by CM metric. The CM value can be calculated using the Vc and Vr values and therefore lead to the categorization of cases. The properties that allow this categorization are showed in Table 1. However, two sub-categories in the spanning cases are determined. *An inter-class spanning case* of a given class is that one, which is partially covered by another case belonging to another class. *An intra-class spanning case* is that one, which is partially covered by another case pertaining to the same class.

It is very important that pivotal cases are kept. Moreover, a representative from of each support case group is kept, the one that has the highest CM value.

Table 1. Properties of the case categories

Type of case	$Vc(ci)$	$Vr(ci)$	$CM(ci)$
Auxiliary case	>1	$= Vc(ci)$	1
Support case group	>1	>1	Same values
Spanning case	≥ 1	>1	≤ 1
Pivotal case	1	1	1

On the contrary, intra-class auxiliary cases contribute nothing in relation to preceding cases and can be deleted. All the same inter-class auxiliary cases with a higher coverage value are kept.

The case-based maintenance algorithm presented in (Haouchine, et al., 2008) consists in determining the categories below according to the CM value and in removing the cases according to their belongings in the classes. This maintenance has needed to be updated by adding new cases. To do this, an auto-increment algorithm has been set up taking into account the structuring and the quality of the case-base resulting after the maintenance step.

2.3 Evaluation

Case-based maintenance algorithm is evaluated on 18 references benchmarks by performance and competence criteria. The training-set contains 80% of cases selected randomly from the case-base and the test-set contains 20% of the case-base. The case-based maintenance algorithm is applied on the training-set according to the test-set. Four columns of Table 2 illustrate the comparison between two methods, CM and ICF relating to the performance. We are particularly interested in the ICF method because it gives the best results compared to the others. It is noted that the CM method is mainly better compared to ICF. Concerning the *storage* in "anneal", "balances-scale" and "mushrooms" databases, the CM is better than the ICF. On the other hand, the accuracy of the ICF is slightly better than that of CM. It means that the large number of cases (nearly double for "balances-scale" and 12.80 for "mushrooms") would compensate the accuracy that is slightly lower in the CM. The same observation is made on the "iris" database. On the other hand, the low rate of competence has been reflected on the low rate of accuracy. By having a low competence rate, the database loses its resolving power (resolution), what is in principle already a disability. This low resolving power was reflected on the accuracy which is decreased about a half. The same observation is made compared to the "glass" database in spite of the good competence rate. Concerning the *storage*, put aside of "anneal", "balances-scale" and "wine" databases, the CM method gave better results than ICF. Especially compared to "iris" with 4.66% against 42.08% and "mushrooms" with 1.05% against 12.80% in which the difference is notable.

Then, the competence calculation will be done for each database according to the CM method. The first made observation related to the competence rate that is higher than 91% for all the databases except of the “iris”. This good competence rate reflects that the case-base maintenance method have reduced the various databases while preserving a good resolution compared to the original bases. However, our method has not succeeded on the “iris” database with 64.00%. Admittedly, there has been a significant reduction but a strong loss of competence. This means that the categorization has not worked well on this type of data.

3. UPDATE OF THE CASE-BASE

After having set up the structuring of the case-base through cases categorization, the aim of the following study is to make evolve the case-base in an incremental way (dynamic) while respecting its structure. This evolution requires the introduction of cases into the case-base under definite conditions. These conditions are directly related to the quality of the case-base. The two criteria allowing to judge this quality were discussed in the section 2.1 namely the competence and the performance. Regarding competence, in our study, the case coverage will concern the problem part and the solution part. Therefore, we will firstly take account of the coverage of the problems part which will be noted (V_{cp}) and secondly the coverage of the solution part which will be noted (V_{cs}). Also concerning the reachability, both notations V_{rp} and V_{rs} will be assigned to the reachability of the problem part and the solution part respectively.

Moreover we associated the *average coverage* of the case-base reflecting the average coverage rate of the entire case-base. This rate will serve as mark for the introduction of a new case in the case-base. The average coverage is a good reference mark of learning cases in the case-base because it characterizes the capacity and the problem resolution space of the case-base. The goal is to increase this space for the case-base in order to cover more problems. Contrary, this increase may seriously damage the structure installation of the case-base, hence the difficulty of the dynamic learning of cases. Consequently, the two steps of case-base maintenance and its auto-increment are highly linked. The average coverage of the case-base is given by the following formula:

$$V_{r_{BC}} = \frac{\sum_{i=0}^n V_{r_i}}{n} \quad (2)$$

Where “n” is the number of cases in the case-base.

This average rate makes it possible to choose the cases which contribute as well as possible to the competence of the case-base. This rate evolves with the evolution of the number of cases obtained in the case-base.

3.1 Auto-increment algorithm of the case-base

When a case is to be added in the case-base, one will add a target case which has been just solved by supplementing its

solution part up to that known, noted: “*target**”. After obtaining the “*target**” we will learn it. To introduce a *target** case in the case-base, it is necessary that the solution proposed by this case does not exist in the case-base. Let the problem part (symptom) be reachable by a number of cases lower than the mean coverage rate of the case-base. This condition will ensure that the introduced case will contribute to the competence of the case-base because its reachability rate will be relatively low.

The incremental learning algorithm is as follows:

```

Let the “CB” Case-Base // CB: Case-Base

Target* case ← Target case // change of the target case
status by associating a solution

For each target* case do

If Vrs > 0 then // the solution of the solved case is reachable
by other solutions of the source cases

    If Vrp < VcBC then // the problem part which is
    reachable by the problem part of the source cases is
    higher than the average coverage rate of the CB

        • Source Case ← Target* case

        • CB ← CB    source case

    EndIf

ElseIf

Cas source ← target* case // Change of target* case status

CB ← CB    target* case // Introduce target* case in the CB

EndIf

EndFor

```

Algorithm 1. Auto-increment algorithm of the case-base

The operating principle of the algorithm is as follows:

According to the solution part reachability rate of the *target** case, it will be allowed or not to be added in the case-base. If “ V_{rs} ” is equal to zero, that means that no similar case with this new solved case was listed before in the case-base, it will be added. On the other hand, if the solution is reachable ($V_{rs} > 0$) then we are interested in the problem part reachability rates compared to the average coverage rate of the case-base. If the reachability rate is less than “ $V_{c_{BC}}$ ” (this means that the covering of the solved case is higher than that of the case-base), then the case is allowed to be learned. This learned case will contribute to improve the coverage rate of the case-base and thus to improve the overall competence.

3.2 Evaluation

To validate the auto-increment algorithm, we propose the following protocol: the case-base will be divided in two parts; the *training-set* contains 80% of case-base and the *new cases-*

set containing the remaining 20% of cases added to 10% of cases selected randomly from training-set. Consequently, the new constituted set will contain 30% of cases from the case-base (Fig. 1). Then, the cases being in a test set are subjected to the training set. If the case in the new case-set fills the necessary conditions to be learned in the training-base then it will be integrated into the latter. And so on for all cases of this set. Thereafter, we obtain the learned case-base containing all the cases from the training-base as well as the cases of the test-base that has met the necessary conditions.

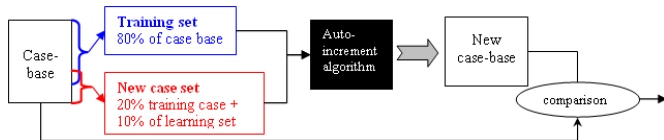


Fig. 1. The protocol set up concerning the evaluation of the auto-increment algorithm

Finally, the rate of the number of learned cases from the structured case-base will be calculated and illustrated in the last column of Table 2.

Table 2. The rate of performance, competence and learning of the various databases relating to CM as well as the comparison with IC

datasets	IC		CM			
	Performance		Performance		Competence (%)	% cases learned in the resulting case base
	Storage (%)	Accuracy (%)	Storage (%)	Accuracy (%)		
anneal	22.59	91.35	29.07	93.85	95.73	100.00
balance-scale	14.67	81.47	30.08	86.16	99.84	81.08
brea-cancer-1	23.51	72.81	11.11	73.66	95.95	92.31
brea-cancer-w	4.27	95.14	2.86	96.78	98.42	100.00
cleveland	15.60	72.08	12.87	78.21	97.35	92.25
credit	16.89	82.28	11.73	84.78	100.00	100.00
glass	31.40	69.64	16.82	43.92	91.58	100.00
hepatits	16.33	82.26	14.83	88.71	95.48	100.00
Iris	42.08	92.56	4.66	48.66	64.00	85.68
lymphography	25.63	77.59	25.00	80.40	98.64	93.78
mushroom	12.80	98.64	1.05	92.70	99.35	88.82
Pima-indians	17.22	69.17	11.06	74.71	99.21	100.00
post-operative	7.18	65.28	3.33	68.22	98.88	98.51
thyroide	21.85	86.63	11.10	62.60	99.80	100.00
voting	8.88	91.19	3.21	91.49	99.31	96.72
waveform	18.98	73.93	17.20	81.26	91.23	97.98
wine	12.00	83.81	22.77	98.97	99.01	100.00
zoo	52.78	92.42	16.83	89.11	96.03	100.00

It is noted that the percentage of the learned cases (the number of learned cases resulting from the auto-increment on the number of initial cases of the case-base) vary from one database to another (from 81% to 100%). This means that the kind of the case is not even a database to another. Concerning the “balances-scale”, “iris” and “mushroom” databases, are databases whose learning rate is lower than 90%. This means that the auto-increment algorithm has detected the cases that were not necessary in the case-base. This is because the maintenance of the case-base was made on cases completely independent of the case-base. As for the databases which

have 100% of learning (“anneal”, “credit”, “glass” ...), it means on one hand that the auto-increment algorithm has performed perfectly by automatically learning all the cases from the test-base.

4. APPLICATION TO SISTRE

SISTRE is a small flexible assembly cell organized into double rings (internal and external). It is composed of five robotized working stations. Each station is equipped with pneumatic actuators (pushers, pullers and indexers) and electric actuators (stoppers) as well as a certain number of inductive sensors (proximity sensors) (Fig. 2). A functional and dysfunctional analysis of this system has allowed building a case-base. It contains 750 cases, 11 attributes and 9 classes. The case problem part contains two kinds of attributes. The attributes reflecting the localization of the pallet and the attributes associated with a value monitored by a sensor, resulting from a supervisor. The solution part provides the repair action associated with the problem.

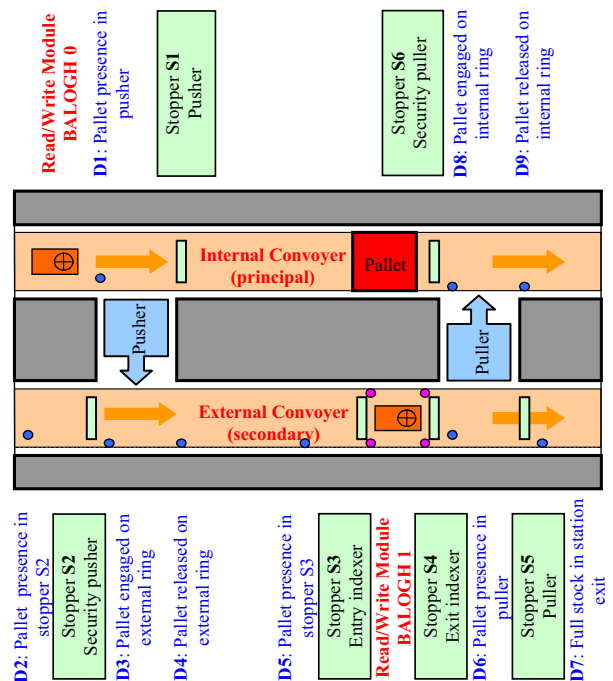


Fig. 2. The working station of SISTRE

We use the same protocol validation discussed in the previous section.

After applying the case-based maintenance algorithm on the SISTRE case-base, the obtained reduced case-base contains 175 cases. A set of 140 cases will represent the training-base and another set of 52 cases will represent the test-base. By applying the principle of incremental learning algorithm, we obtain the 52 cases to be learned, 35 which were added to the training-base and the remaining 17 were considered similar. The results of maintenance and the auto-increment of the case-base are shown in Table 3.

The performance which is translated by the reduction ratio of the case-base according to the accuracy gave a good result, because the obtained reduction is worth more than $\frac{3}{4}$ of the case-base with 100% recognition of classes. As for the competence, which is at 100%, it is shown that the resolution of the obtained reduced case-base is the same as the original case-base.

Table 3. Performance statistics, competence and auto-increment of the SISTRE case-base.

Initial case-base size		750
Size of the obtained case-base		175
Case-base Performance	Reduction ratio	76.67%
	Accuracy	100%
Case-base Competence		100%
Training-base size		140
Test-base size		52
Learning		100%

Concerning the learning, the 25 cases which were added to the training-base, in which there were initially 140 cases, thus they form a total of 175 cases. By comparing the cases of the resulting learned training-base with the reduced case-base, it has been found that there are exactly the same cases and the same number of cases. This gives a learning result at 100%. This result shows that the auto-increment algorithm has performed well by learning only the useful cases in the training case-base. This algorithm allowed to find the initial reduced case-base thanks to an automatic learning process.

5. CONCLUSION

The work approached in this paper is interested in the diagnostic systems using the CBR approach and particularly the maintenance of these systems. To maintain a CBR system consists in maintaining its expertise-base which is the case-base. The case-based maintenance is regarded as an integral part in the CBR cycle. Initially, a proposal for maintenance is done through a case-base structuring method. According to the experiments, the proposed method has provided good results. However, the CBR systems work in incomplete environments in which they evolve through the emergence of new knowledge. Therefore from a partial expertise, a mechanism to update this expertise was installed by proposing an auto-increment algorithm. This algorithm allows enriching the expertise while enclosing new knowledge in the memory in the case form. This proposal for an auto-increment inserts new cases in the case-base while respecting its structuring and quality. A validation protocol of the algorithm was implemented concerning SISTRE (Supervised Industrial System of pallets TRansfer). The results have shown that the algorithm is reliable and that it contributes to improve the quality of the case-base while preserving its structuring. Finally, the two proposed methods for case-based maintenance and the auto-increment are complementary and contribute to the improvement of the CBR system and the evolution of expertise.

REFERENCES

- Aamodt, A. and Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, *AI Communications*, volume (7), page 39-59.
- Aamodt, A. (2004). Knowledge Intensive Case-based Reasoning and Sustained Learning. *9th European Conference on Artificial Intelligence, ECCBR'04*, Lecture Notes in Artificial Intelligence, page1-15.
- Bergmann R., Althoff, K.D., Breen, S., Göker, M., Manago, M., Traphöner, R. and Weiss, S. (2003). Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology. *Springer-Verlag*, 1612, Berlin.
- Brighton, H. and Mellish, C. (2002). On the consistency of information filters for lazy learning algorithms, in: Principles of Data Mining and Knowledge Discovery, *Third European Conference, PKDD '02*, Proceedings.
- Cheetham, W. and Graf, J. (1997). Case-Based Reasoning in Color Matching. *The 2nd International Conference on CBR R and D*, volume (1266) page1-12.
- Devaney, M. and Cheetham., B. (2005). Case-Based Reasoning for Gas Turbine Diagnostics. *In 18th International FLAIRS Conference (FLAIRS-05)*.
- Grant, P.W., Harris, P.M., and Moseley, L.G. (1996). Fault Diagnosis for Industrial Printers Using Case-Based Reasoning. *Engineering Applications of Artificial Intelligence*, volume (9), page163-173.
- Haouchine, K., Chebel-Morello, B. and Zerhouni, N. (2008). Competence-Preserving Case-Deletion Strategy for Case-base Maintenance. *Uncertainty, Similarity, and Knowledge Discovery in Case-Based Reasoning workshop. 9th European Conference on CBR*.
- Iglezakis, I. and Roth-Berghofer, T. (2000). A survey regarding the central role of the case-base for maintenance in case-based reasoning. *In Mirjam Minor, editor, ECAI Workshop Notes*, page 22-28.
- Leake, D.B. and Wilson, D.C. (1998). Categorizing case-base maintenance: dimensions and directions. *4th European Workshop on Case-Based Reasoning '98*, page 196-207.
- Leake, D.B. and Wilson, D.C. (2000). Remembering Why To Remember: Performance-guided casebase maintenance. *Advances in Case-Based Reasoning: EWCBR-2K*.
- Smyth, B. and Keane, M.T. (1995). Remembering To Forget: A competence Preserving Deletion Policy for Case-Based Reasoning Systems. *In Proceeding of the 14th International Joint Conference on Artificial Intelligent, Morgan-Kaufmann*, page 377-382.
- Smyth, B., and McKenna, E. (1998). Modelling the Competence of Case-Bases, *Advances in casebased reasoning, Lecture notes in computer science*, volume (1488), page 208-220.
- Smyth, B. and McKenna, E. (1999). Building Compact Competent Case-Bases. *Case-based reasoning research and development, Lecture notes in computer science*, volume (1650), page 329-342.
- Smyth, B. and McKenna, E. (2002). Competence models and the maintenance problem. *Computational Intelligence: Special Issue on Maintaining CBR Systems*, In Press.
- Yang, Q. and Zhu, J. (2001). A case addition policy for case-base maintenance. *Computational Intelligence Journal, A Special Issue on Case-Base Maintenance*, volume (17), page 250-262.