Case Adaptation for Diagnosis in Manufacturing

B. Radhika Selvamani and Deepak Khemani

Artificial Intelligence and Data Bases Lab
IITMadras,India
bradhika@cse.iitm.ernet.in,khemani@iitm.ac.in

http://aidb.cs.iitm.ernet.in

Abstract. In complex domains with poor domain knowledge it is possible to learn adaptation knowledge from the cases. This paper addresses the problem of adaptation for case based diagnosis in manufacturing domain. Diagnosis in manufacturing deals with tuning a few parameters, which would eradicate the possible defects in the products. Each case stores a product manufacturing experience and the outcome of its application in the domain. Whenever a defect occurs in the domain, the case base is consulted for a similar case, which has been rectified earlier in the domain. Initially it was assumed that experts would provide remedies for each of the defective cases. Knowledge acquisition turned out to be a problem. Hence there is a necessity to discover the cause for the defects and to adapt the defective cases accordingly. To integrate the discovered knowledge with the CBR cycle we propose a critic based adaptation approach. Given a defective product the problem is to select the relevant parameters and tune them accordingly. Real time experiments are costly to evaluate the rectified cases. We have designed an experimental setup, which captures the behavior of the adapted cases.

Key words: Case Based Reasoning, Adaptation, Decision Trees, Utility

1 Introduction

Diagnosis using CBR in medical domains consists of identifying the disease and suggesting therapy for the same given the observed symptoms. Fault diagnosis in electric circuits is arriving at the faulty component given the circuit, input and the output. Diagnosis in manufacturing deals with tuning a few parameters, which would eradicate the possible defects in the products. Knowledge acquisition for diagnosis is a difficult exercise. We have proposed a method to integrate the induced diagnosis knowledge in the adaptation cycle of the CBR system.

1.1 Related Work

A very informative classification of adaptation techniques and a statistic based on how frequently have they been used has been reported by Hanneay et.al[1]. Yet there has been no known uniform approach to tackle the adaptation itself. Nevertheless

Wilke et.al[2] has established a framework for knowledge light approaches to acquire adaptation knowledge from knowledge containers, which provides some insight to view the problem of adaptation in general. Our approach is a kind of critic-based adaptation where the feedback of applying a case decides the kind of adaptation to be performed. With respect to the latest taxonomy provided by hanney we place the kind of adaptation we perform in single case, compound manipulable solution with interaction among the solution attributes. The related work in configuration and design [3] has compound manipulable solutions with less interaction among the components. Hence it was possible to adapt the numerical and nominal components separately whereas we had to consider the interaction among the solution attributes while eliminating the defect. The recent contributions in adapting single and multi dimensional solutions using statistical learning methods, and search algorithms have been discussed by Pal et.al.[4]. Most of these methods need the problem to be formalized in to case pair differences, which may not be possible in case of nominal attributes which interact with numerical attributes. Moreover they focus more on enhancing case coverage and do not suit a critic based adaptation scenario. Though we use decision trees to obtain the patterns, which would be used in adaptation, the way we create the training samples and the way we adapt the cases using the decision trees are quite different from the earlier approaches. Hence we choose to report our case of adaptation as a unique method of adaptation suiting specifically for diagnosis in manufacturing domains.
In the following section we describe the case based reasoner working in the manufacturing domains and their tasks. Then we proceed describing the need for adaptation. Then we describe the way we learn the patterns, which are used to adapt the cases. We describe the set of experiments performed to observe the behavior of the adapted cases. We also compare another way of solving the problem without adaptation called utility-based retrieval and compare our results with adaptation.

2 CBR in Manufacturing

The special focus in manufacturing domain is a result of a series of CBR projects[5] we did in manufacturing domains as a knowledge management exercise[6]. In our feasibility study it turned out that diagnosis of defects is the costliest knowledge operation in terms of time and money. Our case vocabulary in each of these domains was chosen to suit the diagnosis task in particular, but could also be configured for design processes with more accumulated cases. We chose the case vocabulary to include almost all parameters involved in manufacturing a product from the design specifications to the process parameters along with the final inspection and outcome about whether the product was dispatched or not.
Initially the assumption was that the experts would provide the remedy for the defects while authoring the case. But acquiring rectification knowledge from the experts was not possible since the diagnosis data was not captured at the operational level. The defect rectification knowledge was dispersed with the shop floor personnel and was mostly tacit and adhoc. Instead of expecting a process change in the organization to capture the defect rectification data, which existed with different shop floor personnel

we proceeded towards knowledge discovery from the accumulated good and bad experiences. The experts were satisfied with the decision trees but were unable to use it as such. We realized the need to integrate the knowledge discovered with the CBR system and this paved way for the whole process of critic based adaptation for diagnosis in manufacturing.

2.1 Domain and Case Vocabulary

Refractory blocks manufacturing domain, steel strips manufacturing domain and steel tubes manufacturing domain are the three domains where the CBR systems have been implemented. Though the products are very different, we use the same case structure and a common CBR kernel in all the three domains. The product specification, the various process parameters, the defect descriptions and the rectifications for each product are compiled into a single case in the case base. There are two phases in which the CBR system functions based on what part of the case forms the problem. For a diagnosis phase the rectification forms the solution part and the rest of the attributes form the problem part. Alternatively for a design phase the product specification becomes a problem part and the rest becomes the solution part. In the refractory blocks domain the foundry department decided the characteristics of the mould to be used whereas the furnace department decided the furnace parameters for the same product specification. If a defect occurs in a product the experts are unable to relate it to the pattern of process components, which could have caused it. This creates a gap in filling up the remedy part in each defective case.

3 Case Adaptation

The CBR kernel performs case retrieval by a linear search over the case base using a kNN approach. The similarity metric is the weighted sum of the local similarity measures for all the problem attributes. We use a flat case structure because we acquire most of the case details from existing databases. Also we included about 200 attributes in each domain, which is quite large compared to earlier CBR systems. The CBR system is by default at the diagnosis phase. During the seeding stage every product-manufacturing instance whether it is good or bad is added to the case base with or without remedy. Whenever a defect occurs in a product the case base is sought for a remedy. If the closest matching past case does not provide a remedy the retrieved case is adapted in a design phase.

3.1 Case Adaptation Pattern

The adaptation patterns are the combinations of process parameters, which would affect the final outcome of a product. A particular process pattern when observed in a product suggests a good or bad effect in the final product unlike the traditional adaptation rules, which suggest the amount of change to be made to the final solution. Let us visualize the concept of patterns and adaptation in an example metric case space (Fig. 1). Each case has four attributes, a problem attribute, two solution attributes and an outcome about whether it has a particular defect or not. The outcome may be visualized by the shape of the cases. Let the filled square represent the query. It is in the defective partition and the goal is to adapt it to the nearest good partition as shown. A decision tree has been learnt for the cases. The nodes of the tree correspond to a partition in the case space. The corresponding patterns have been tabulated (Table 1). The patterns have been scored according to their pattern selection score (Info) and similarity(Sim) to the query. The query is then adapted according to the highest scored pattern ($s2 \leq 5$) from (9,6) to (9,4.5). The pattern ($s2 > 5$ and $s1 \leq 6$ and $s1 > 4$) would adapt the query (9,6) to (5.5,6) which is less similar to the query and hence not desirable.

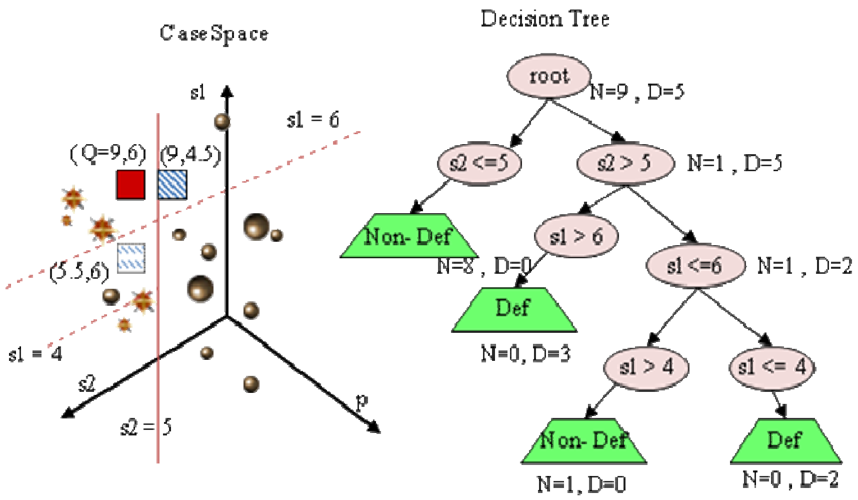

Fig. 1. Metric Case Space and Decision Tree.

Table 1. Patterns obtained from the decision tree with scores.

	Patterns	Siblings	Sim	Info	Score(θ) Sim*Info
1	s2 <= 5	s2 > 5	0.90	0.70	0.63
2	s2 >5 & s1> 6	s2 > 5& s1<=6	0.60	-0.29	-0.17
3	s1>5 & s1<=6 & s1 > 4	s1>5 & s1<=6 & s1 <= 4	0.60	1	0.6
4	s2 > 5	s2 <= 5	0.90	-0.70	-0.42
5	s2>5 & s1<=6	s2 > 5 & s1> 6	0.60	-0.29	-0.18
6	s1>5 & s1<=6 & s1 <= 4	s1>5 & s1<=6 & s1 > 4	0.60	-1	-0.6

3.2 Pattern Selection Score

We learn a separate set of patterns for each defect. The patterns for a particular defect are obtained from the process instances as follows. We label the process instances D-Defective and N-Non-Defective based on the defect to be learnt. We learn a decision tree from the training data. Each node of the decision tree corresponds to a pattern p (Fig.1)and (Table.1). The sibling nodes correspond to a pattern s. The (p,s) pair decides the importance of the pattern p in rectifying the defect. The pattern selection score has been described in Table.2.

Table 2. Pattern Selection Score

Relevance Score	Formulae
InfoGain	$\dfrac{I(Par) - \dfrac{Dp+Np}{Dpar+Npar} I(P) - \dfrac{Ds+Ns}{Dpar+Npar} I(S)}{I(Par)}$ Score = - InfoGain when $D_P > N_P$
Total Defects : D Non-Defects : N Node Par Dpar Npar Node P Node S D_P N_P D_S N_S	$I(P) = -\left[\dfrac{Dp}{Dp+Np}\log\dfrac{Dp}{Dp+Np} + \dfrac{Np}{Dp+Np}\log\dfrac{Np}{Dp+Np}\right]$ $I(S) = -\left[\dfrac{Ds}{Ds+Ns}\log\dfrac{Ds}{Ds+Ns} + \dfrac{Ns}{Ds+Ns}\log\dfrac{Ns}{Ds+Ns}\right]$ $I(Par) = -\left[\dfrac{Dpar+Dpar}{Dpar+Npar}\log\dfrac{Dpar}{Dpar+Npar} + \dfrac{Npar}{Dpar+Npar}\log\dfrac{Npar}{Dpar+Npar}\right]$

4 Experiments

The primary objective of the experiments carried out was to determine the performance of adapted cases in the domain. One of the ways to measure the performance of the defective cases is to manufacture the product with the rectified parameters. But this would be a costly experiment and it is difficult to realize this task in an organization, which has standard manufacturing procedures. So we decided to use the past data available in the domain to study the behavior of adapted cases. We assume a case as a cluster of similar manufacturing instances in the domain. The final outcome of a product is probabilistic in nature. The utility of a case may be determined by the probability of successful member instances it holds. We build a feedback case base where success probability of a case gets updated whenever an instance is augmented to it. The past data collected in the domains had both good and bad manufacturing instances. We collected a year's data in the three manufacturing domains for a selected set of products. We identified key defects in the products that need to be diagnosed. We created 12 datasets in all three domains for each manufacturing department labeled according to the defect to be diagnosed. For each dataset we created a test bed as follows.

– We randomly picked up a few defective instances for testing.
– We used the rest of the data
  - to train a decision tree using c4.5 [7] algorithm for adaptation
  - to build a feedback case base for predicting the outcome of a process instance in the domain.

In practice a defective case retrieved by a CBR system would be adapted by a decision tree. But for the purpose of evaluating the adaptation process, we preselect the defective instances and adapt them directly using the decision tree and evaluate them using the feedback case base. The outcome of the defective instance before and after adaptation is obtained by observing the utility of the nearest matching case in the feedback case base. Building the feedback case-base is same as building the seed case base with an additional functionality of maintaining the success probability of each case.

4.1 Increase in Success Probability after Adaptation

To observe the success probability of the defective instances after adaptation we plot the average success probability of the closest matching cases before and after adaptation of the defective instances picked up in each dataset (Fig. 2).

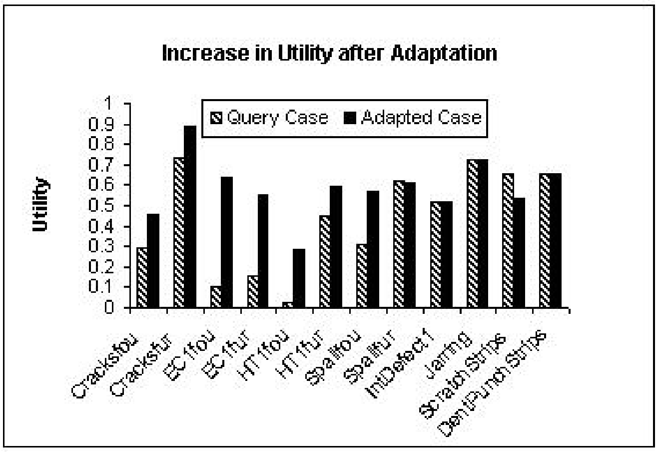

Fig. 2. Increase in Utility of the Adapted Cases.

4.2 Successful and Failure Adaptations

To observe the number of cases adapted successfully (hikes) and those, which were adapted badly (drops) we plot the total hikes and drops in all datasets for different augmentation thresholds of the feedback case base (Fig. 3). We can also see there are a number of instances which retrieved the same case after adaptation (nChange). This may be because there is no considerable change after adaptation or because there are no cases in the case base, which capture the true behavior of the adapted cases.

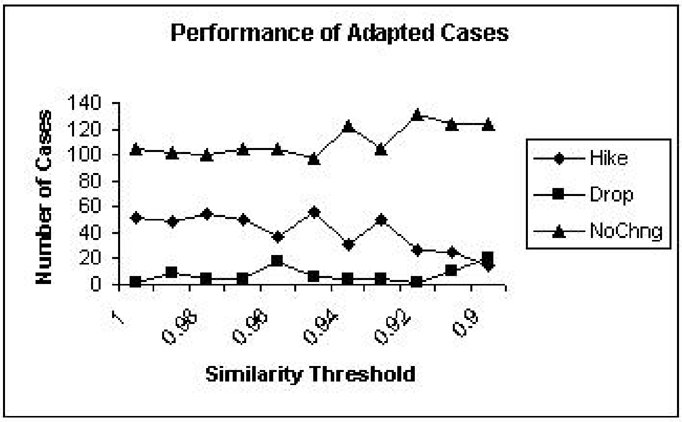

Fig. 3. Number of successful adaptations (Hikes) and failures (drops)

4.3 Utility Based Retrieval

We have used a utility score based on the similarity and success probability of the case to the query for retrieval. Results have been plotted in Fig. 4.
Utility based retrieval score (case,query)
= Similarity(case, query) x Success probability (case)
Utility-based retrieval gave the highest success probability compared to adapted cases. But the similarity with the query is considerably less compared to the adapted cases. Also we could observe that utility-based retrieval may alter attributes irrelevant to the defect whereas decision tree based adaptation strictly adapts only the relevant attributes.

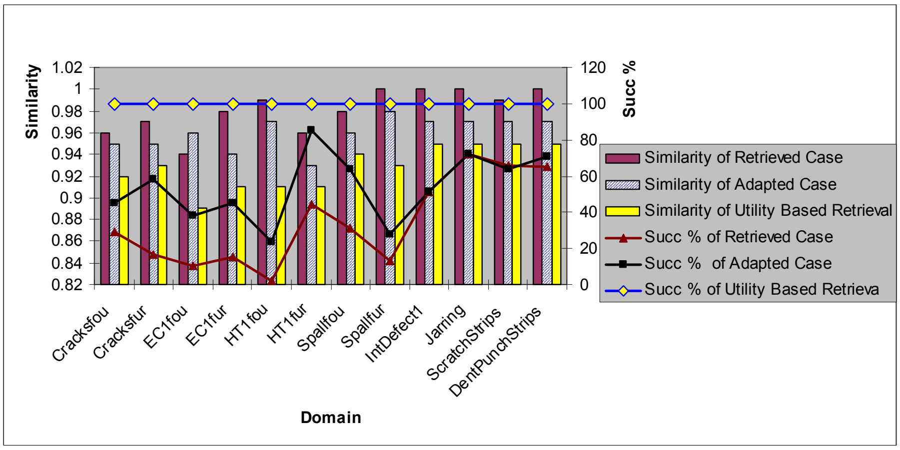

Fig. 4. Utility Based Retrieval Vs Adapted Similarity Based Retrieval

5 Conclusions

We have discussed the problem of diagnosis in a manufacturing setting and the practical difficulties in realizing a purely retrieval based CBR system. We have suggested the use of critic-based adaptation where the adaptation knowledge is learnt from the past product manufacturing history. We have designed an experimental setting to observe the performance of adapted cases using a feedback case base since it is difficult to perform real-time experiments in the domain. We have plotted the observed results. Retrieval-based approach is suitable only when all the defective cases in the seed case base have been diagnosed earlier in the domain. When not all cases have been diagnosed it is possible to learn the diagnosis knowledge from the accumulated good and bad experiences in the domain. This knowledge is used to adapt the retrieved defective case for a query. Utility-based retrieval may complement the adaptation-based approach for better performance.

Acknowledgments. We acknowledge the support of Murugappa Group of Companies Ltd., Chennai in funding the research projects as well as providing required data for the experiments.

References

1. Hanney, K., Keane, M.T., Smith, B., Cunningham, P.: Systems, tasks and adaptational knowledge: Revealing some revealing dependencies. In: In proceedings of the 1st International Conference on Case based Reasoning, Portugal, Lecture Notes in Artificial Intelligence, 1010, Springers Verlag (1995) 461-470
2. Wolfgang Wilke, Ivo Vollrath, K.D.A.: A framework for learning adaptation knowledge based on knowledge light approaches. In: In proceedings of the 5th German Workshop on CBR. (1997)
3. Jarmulak, J., Craw, S.: Using case-base data to learn adaptation knowledge for design. In: In proceedings of the seventeenth International Joint Conference in Artificial Intelligence, Morgan Kaufmann (2001) 1011-1016
4. Pal, S.K., Shiu, S.C.: Foundations of Soft Case-Based Reasoning. John Wiley and Sons, Inc., New Jersey (2004)
5. Khemani, D., Selvamani, R.B., Dhar, A.R., Michael, S.M.: Infofrax: Cbr in fused cast refractory manufacture. In: In Advances in Case Based Reasoning 6th European Conference ECCBR06, Aberdeen, Scotland, Springer Verlag (Sep 2002)
6. Selvamani, B.R., Khemani, D.: Layers of memory and reasoning in the problem of diagnosis in manufacturing. In proceedings of the Workshop on KMOM colocated with IJCAI07, Hyderabad, India. (Jan 2007)
7. Quinlan, J.R.: C4.5 Programs for Machine Learning. Morgan Kaufmann (1992)