Efficient Condition Monitoring and Diagnosis Using a Case-Based Experience Sharing System. Mobyen Uddin Ahmed, Erik Olsson, Peter Funk, Ning Xiong, in the 20th International Congress and Exhibition on Condition Monitoring and Diagnostics Engineering Management, p 305-314, COMADEM 2007, Faro, Portugal, June, 2007

EFFICIENT CONDITION MONITORING AND DIAGNOSIS USING A CASE-BASED EXPERIENCE SHARING SYSTEM

Mobyen Ahmed, Erik Olsson, Peter Funk, Ning Xiong

Department of Computer Science and Electronics, Mälardalen University,
PO Box 883 SE-721 23, Västerås, Sweden
{firstname.lastname}@mdh.se
Tel: +46 21 10 14 53, Fax: +46 21 10 14 60

ABSTRACT

In a dynamic industrial environment changes occur more and more rapidly, new machines, new staff when scaling up production and reduced staff when scaling down during a recession, staff with varying experience etc. This puts a high focus on experience reuse and sharing; much experience is lost during down-scaling and tied up in knowledge transfer/teaching during up-scaling. This is recognised as very costly for industry and reduces productivity and competitiveness. Condition Monitoring and diagnostics is such an area where lack on knowledge and mistakes can have severe consequences for a company's long term existence. Maintenance staffs, technicians and engineers also gain much experience during their every day work, often during many years, but there are rarely any good processes for experience sharing and reuse inside the organisations. In this paper we present an experience sharing system based on case-based reasoning and limited natural language processing. The system is a tool for maintenance staff and engineers and enables efficient experience collection, reuse and sharing. The implemented prototype is web-based to promote access from any location and may be local or global enabling experience sharing openly or in clusters of collaborating companies. Case based reasoning has proven to be an efficient method to identify and reuse experience if the application domain has cases. Our target application domain has these features and there are plenty of cases valuable to reuse. We have validated this in close collaboration with maintenance engineers through field studies. The prototype developed shows promising features and will be tested in real industrial environments during 2007 and 2008.

KEYWORDS

Condition Monitoring, diagnostics, experience, knowledge, sharing, reuse, Case-Based-Reasoning, Natural Language Processing.

1. INTRODUCTION

A skilled engineer with long experience can save a company large amounts of time and money, e.g. repairing a machine the first time may take a day or two, but if carried out the second time by the same team or with someone who has done it before may reduce the time to a few hours. It is obvious that industry can make large savings by introducing tools and routines for more efficient experience reuse and the benefit of experience sharing and reuse has become obvious with the increased use of internet and the potential of large economical benefits is obvious. Today almost all over the world people in different fields are taking advantage of sharing their experiences and reusing them for example in Knowledge Management (Delaître and Moisan, 2000; Marjanovic, 2005), Diagnostics and Condition Based Maintenance system (Funk and Jackson, 2005)and Health Monitoring system (Funk et al. 2006). At the same time the amount of low quality information and waste amount of information available reduces the value of the internet and the time employees spend on searching for the right information increases and it may extend to hours every day (Blomberg, 2005). Also much of the information and experience available is of low quality and may even be wrong and in worst case resulting in serious accidents and costs.

Industries today have increasingly smart diagnostic systems or monitoring systems. This shifts the focus on the next step, i.e. how to resolve the problems. This is often one of the most crucial steps with an largest impact on costs and time. Different circumstances may need different solutions for a problem. A standard solution, e.g. stopping the production and replacing some parts may be very costly. There may be experience available how to amend the problem temporarily to respond to e.g. a time critical customer order. Identifying past experience relevant for the situation will help the engineers to take a better more informed decision and avoiding mistakes. In many engineering environments a domain dependent experience sharing system where experience is gathered, stored and reused efficiently would be a valuable tool for engineers. This would increase productivity and efficiency and promote experience transfer between engineers.

For example if the monitoring system notifies a deviation in a machine (Funk and Jackson, 2005), a fault report is often written; an engineer makes a diagnosis and may order spare parts to repair the machine, carries out the replacement, tests if it corrected the problem and after that restarts the production. These fault report; spare parts, required time and statistics on production and performance after repair are also stored in often different databases but so far they are rarely systematically reused. Many companies and industries today have a large untapped potential of experience reuse and would benefit from a more systematic approach towards knowledge and experience reuse.

Case-based reasoning (CBR) is a methodology for problem solving reusing previous experience (Watson, 1999) and also for collecting new experience since every new problem case, once solved, becomes a new case that may be stored and reused. In this paper we propose a Case-Based experience sharing solution that enables reuse of experience in a more efficient way compared with what is common practice in industry today. The system identifies and presents the most significant experiences to assess from the collaborative space where experiences from various companies may have been stored under many years. It may work globally through the internet and gather and share textual experiences but it can also use structured experience and mixed representations with both textual and non-textual features. (Wilson and Bradshaw, 2000)

For these textual cases the *tf-idf* (term frequency–inverse document frequency) (Salton and Buckley, 1987) weighting scheme was used in the vector space model (Salton et al., 1975) together with cosine similarity to determine the similarity between two cases (Rosina et al., 2005). Additional domain information often improves results, e.g. a list of words and their synonyms or dictionaries that provide comparable words (Scott and Matwin, 1998; Recio et al., 2005) and relationships within the words using class and subclass (Mladenic and Grobelnik, 1998). Our proposed system uses domain specific ontology that represents specific knowledge (Swartout W, 1999; Gruber, 2006) i.e. relation between words. In the domain of industrial robots, an ontology with entities and relationships could be found in the form such as, ‘axle’ (is an arm) is <a part of> ‘gearbox’ (a unit of a robot). By using ontology the identification of similar cases improves if words in the ontology are used in the cases.

Such a CBR based tool enables experience gathering, sharing and reuse in e.g. production industries by facilitating the users with an interactive tool. Since the cases have authors the system also helps in identifying the right person, e.g. there may be an engineer or operator near by and available for assistance, this would be an ultimate solution. Depending on the user and their security level; system will allow sharing knowledge and reusing experience among the collaborating companies. Reusing experience will not only shorten the time needed to solve an approaching problem, it also enables avoidance of expensive mistakes which will increase the participating companies' competitiveness.

2. SYSTEM OVERVIEW

An illustration of how the system may work in an industrial context assisting production planning is given in figure 1 where the user has the task to make a decision, although there is a good fault diagnosis system. The proposed system can find a solution that can help the company to run the production and at the same time both companies can share the knowledge if they collaborate with each other.

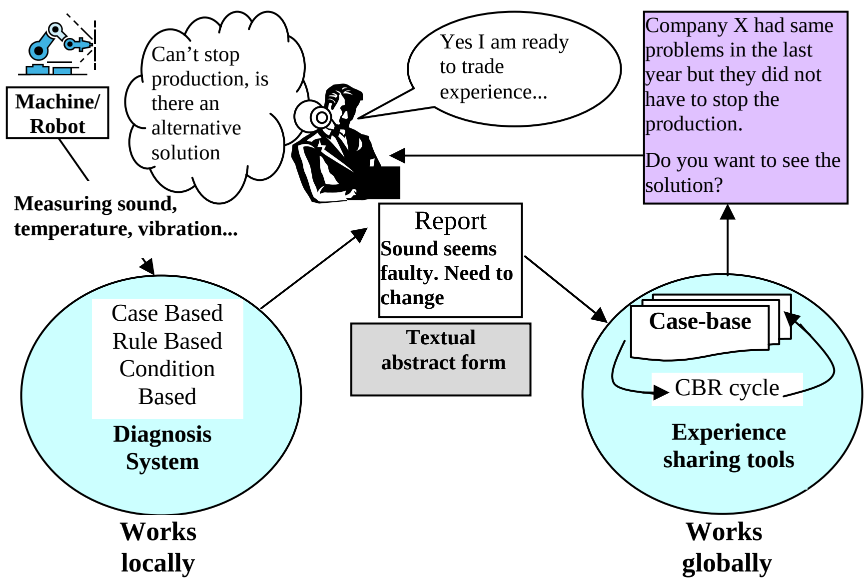

Figure 1 Example of how the system may be used in an industrial context

3. WHY CASE-BASED REASONING?

Human solve problems by using both his own experience as well as that learned from other experienced people, simulations, modelling etc. It is always valuable with a second option and a system able to identify similar and relevant past cases is a tool that technicians would appreciate (Blomberg, A. 2005). The methodology of Case-Based Reasoning (CBR) is used to solve new problems often by using existing experience that is obtained by remembering a previous similar situation.

CBR (Aamodt and Plaza, 1994; Watson,. 1997) is a method based on learning from similar cases stored in a case library that is a plausible cognitive model of some human problem solving. A CBR cycle with 4 steps as shown in the figure 2: Retrieve, Reuse, Revise and Retain has been introduced by Aamodt and Plaza (Aamodt and Plaza, 1994).

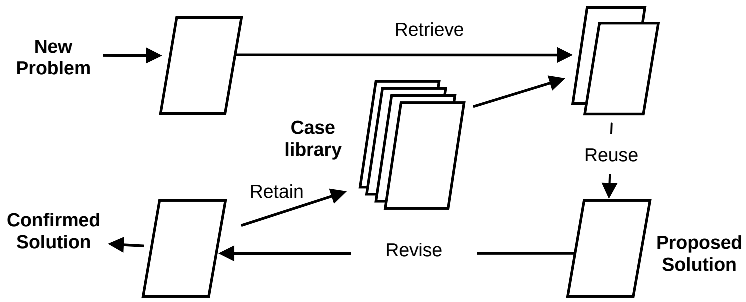

Figure 2 The case-based reasoning cycle as proposed by Amodt and Plaza (Aamodt and Plaza, 1994).

In the retrieve step, a new query for a specific problem is posted and the system tries to retrieve a set of similar cases by matching the query against the previous cases from the case library. Domain knowledge, e.g. ontology is used in matching to identify similar cases and commonly the nearest neighbour search is used to identify similar cases. If it finds any suitable case that is sufficiently close to the current problem then the solution is reused (after some adaptation and revision if necessary). The user may revise the selected case and retain this new solution, its outcome, time used etc. along with the initial problem description into the case library.

3.1 Case Structure

A case structure can contain a contextualized piece of knowledge that represents an experience. The case typically consists of a problem specification and solution where we can store most types of data such as textual values (e.g. names, addresses), numeric values (e.g. cost, ages) and multimedia features (e.g. photographs, sound, and video). An important issue is to find a suitable structure and features for a specific domain.

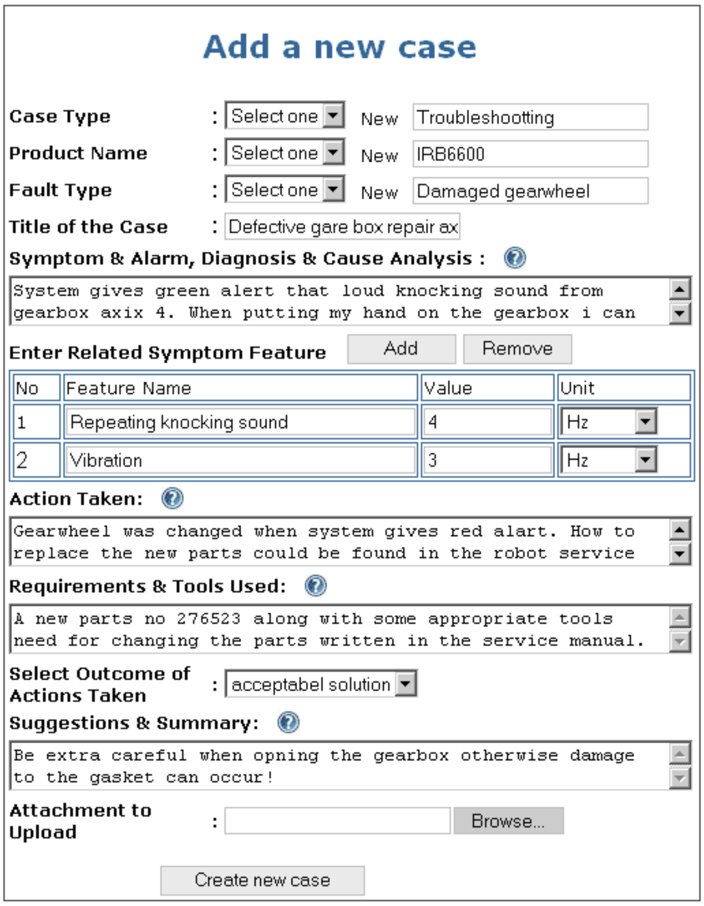

Figure 3 General case structure and its presentation

If no important features are extracted and if the structure is too ambiguous the result of the matching will be less good. In the industrial domain, knowledge is often stored and described with free or semi-structured text instead of some predefined structure but in our system cases are represented as a combination of both unstructured text and structured data. A sample screen shot for the case structure is shown in fig 3.

The problem description is represented both in textual format as well as with a number of features. In figure 3 users can enter text to describe symptoms, diagnosis and cause analysis with alarm; in this representation user can also chose the outcome and can indicate the success rate.

3. 2 Case Retrieval

The CBR systems include the essential steps such as retrieval, reuse, revise, and retain. The retrieval step is the most important step where the aim is to find the most similar case(s) which may be reused. The procedure of case retrieval begins with identifying the most important features, then does some search and matches, and ends up with selecting the most similar case(s). The different steps in the retrieval of similar case(s) are shown in fig 4.

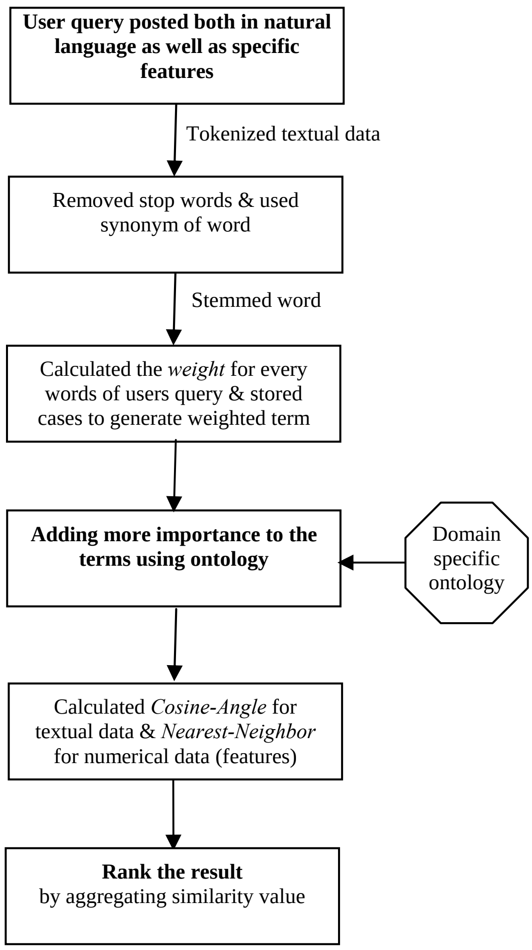

Figure 4 Different steps for case retrieval.

User's specific problem description can be given into the system through the user interface both in a natural language format as well as in form of specific features. The text tokenizer algorithm decomposes the whole textual information into sentences, and then into individual words. Because of the huge amount of words, a filtering step is needed to improve retrieval effectiveness. Three techniques were used; firstly removing stop-words and special characters by the stop-words and special characters blacklist for both user query and stored cases. Then, a list of synonyms of the words were also used to reduce the number of terms and Porter stemming algorithm helps to stem the words that provide the ways of finding morphological variants of search term. After calculating the weight for each word, words are represented as terms in a vector space and by using domain specific ontology we could improve the importance for candidate terms before measuring the cosine angle for the textual information among the stored case and user query case. Different features values along with local weights were used to find similarity between the features of stored cases and the features of new case where Nearest-Neighbour algorithm works perfectly. For presenting the sorted results we aggregated all local similarity values both for user query and a stored case, which is described in the following section.

3.2.1 Weighting Terms (Wi,j)

From the different algorithms for calculating the weight of each term among the stored cases and the input query (such as $W_{i,j}$ method, Relating term precision to term frequency method, Term discrimination method, etc), the weighting terms method (Garcia, 2006) was chosen to perform the further matching. The general equation for $W_{i,j}$ can be shown by equation (1).

$$W_{i,j} = tf_{i,j} * idf_j = tf_{i,j} * \log_2(\frac{N}{df_i}) \qquad \textbf{(1)}$$

Where, $W_{i,j}$ is the weight of term T_j in the case C_i, $tf_{i,j}$ is the normalized frequency of term T_j in the case C_i and idf_j is the inverse case frequency where N is the number of cases in the database and dfi is the number of cases where term T_j occurs at least once.

Cases in the database have been processed according to term vector model and stored in a separate table when new cases were stored. First, an index of terms from the case collection is constructed; the frequencies of the terms ($tf_{i,j}$) appearing in each case (C_i) and new case query (Q) are counted. Then, the cases (collection) frequency (df_i) and idf_j calculations are straightforward. Finally, the $tf_{i,j}$* idf_j product gives the weight for each term.

3.2.2 Enhanced Term Vector using ontology

When we already calculated the weight of each term for every case where terms of each case are satisfied with other case by exact match or synonym or having a co-occurrence; but still some words have a complex relationship that can be defined by ontology (i.e. the term axel and gearbox) could increase the weight of that case. We could improve the weight to the vector term as:

1. If a term T_f in the case is related to a term T_o in the ontology but the term (T_o) does not exists in the case then this term could be added as a *new* term by same importance as the source terms of weight that is the score of *tf-idf.*
2. If a term T_f in the case is related to a term T_o in the ontology and also the term T_o is exists in the case then there relatedness and strength of term could be added to the source terms of weight that is score of *tf-idf.*
3. If there is more than one term in the case related to one term in the ontology then those terms will get more importance for that case by adding their relationship strength.
4. If there is one term in the case related to more than one term in the ontology then normalized strength of their relationship could be added to the source term.

An example is shown in fig.5 on how the ontology helps to improve the weight vector.

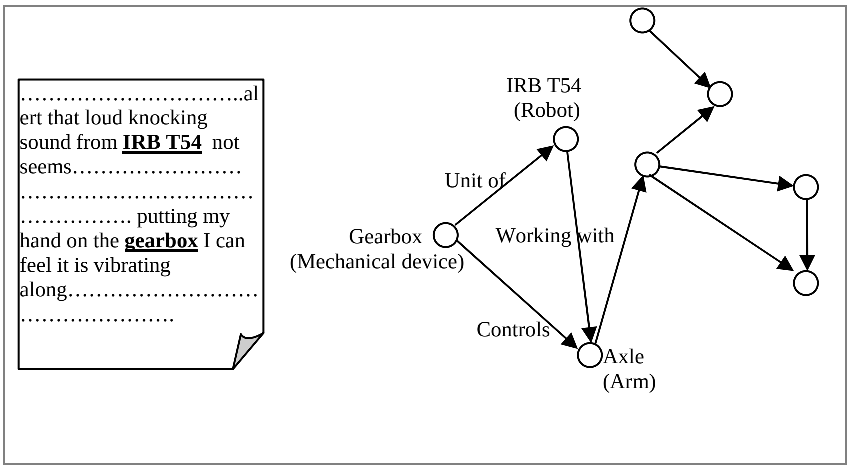

Figure 5 Weighting term vector using domain specific ontology

From the figure 5 "axle" is a term in the ontology related with the term "gearbox" which is in the case text, so term "axel" is also important for this case and could be added. The terms "IRB T54" and "gearbox" are already in the case and have a relation in the ontology so it will increase the importance of this case. Again the terms "IRB T54" and "gearbox" are related with another term "axle" in the ontology so it will be more important term for this case. Thus the terms get importance depending on ontology and will make a more efficient calculation in similarity matching.

3.2.3 Similarity Functions

To find the similarity between stored case vector C_i and new case query vector Q we implemented cosine similarity function (Garcia, 2006) for the textual information. This ratio defines the cosine angle between the vectors, with values between 0 and 1 and can be calculated by the equation 2.

$$Cos\ \theta_{C_i} = Sim\ (Q, C_i) = \frac{Q \bullet C_i}{\|Q\| \|C_i\|} = \frac{\sum_i w_{q,j} w_{i,j}}{\sqrt{\sum_i w_{q,j}^2}\sqrt{\sum_i w_{i,j}^2}} \qquad \textbf{(2)}$$

Where $Cos\theta$ is the cosine angle between each case and query case could be defined by the similarity function $Sim\ (Q, C_i)$. For every stored case dot product was calculated with the query case by $Q.C_i$ where zero products ignored, next vector lengths were calculated for every case and query case (zero terms also ignored) where W are weights calculated through equation 1.

For the numeric value such as specific features both in user query and stored cases K Nearest Neighbor (K-NN) along with Euclidian distance algorithm work well, the equation of these function are described below:

$$Similarity\,(T,S) = \sum_{f=1}^{n} W_f * sim(T_f, S_f) \qquad \textbf{(3)}$$

In the equation 3, *Similarity(T, S)* calculate global similarity for all the numeric features, T is the query/current case, S is the case from source, W_f is the normalized weight defined by each local weight (given by the user) divided by sum of local weights of individual features f , n is the number of the features in each case, f is the individual feature from 1 to n, and *sim* is the local similarity function for feature in case T and S (adopt Euclidean distance) as described in the eq. 4.

$$sim(T_f, S_f) = 1 - \frac{abs(T_f - S_f)}{Max(T_f, S_f) - Min(T_f, S_f)} \quad (4)$$

$Sim(T_f, S_f)$ represents local similarity for each feature, function *abs* is used to get the absolute values and *Max* and *Min* of the features values are derived from the whole case base and user query.

4. RESULT

The system formulates a ranking of stored cases based on the cosine angle value for each case paired with the problem description. All the cases are listed according to the percentage where 100% means the perfect match of all relevant features both extracted from the text as well as features from the structured data. Candidate and/or acceptable similar cases were presented along with type of solution and its success rate. The problem, solution and its related parts can be presented according to user profile. For instance, if a matching case is unrestricted or local then the solution is shown, otherwise the user has to contact the owner of the case to get access to the solution. A screen shot for a search is shown in fig 6.

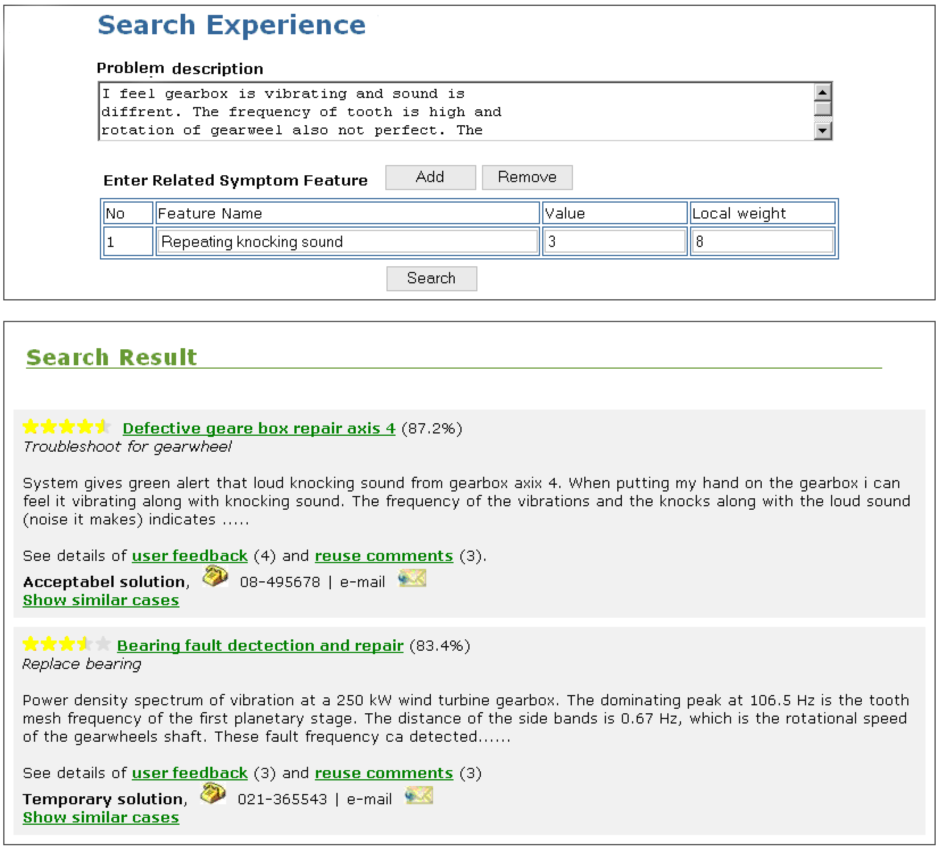

Figure 6 Example of an experience search for a solution to a problem.

The user interface for searching experience is consisting of a text area along with textbox where user can enter textual query and the threshold value in percentage. After finishing the search, depending on threshold value the most relevant case(s) will be ranked with case title and type along with the score (similarity value). User can also get the information about success rate, feedback and comments from other users. In the reuse step the retrieved cases are reused to solve a given new problem. In our system user can give feedback and comments about the case(s); for example user will rate for a retrieved case in terms of how much the case has been matched with his/her current problem and how valuable the case is. This information will help the system both in evaluating matching algorithm and case indexing.

5. SUMMARY AND CONCLUSION

We have shown how case-based experience reuse is able to reduce costs in an industrial environment by transferring relevant knowledge to an engineer that has a problem to solve. It not only identifies valuable experience for the current situation, it also enables avoiding repetition of sometimes very expensive mistakes. Since cases also have case authors and may have references to what experience different technicians have in this particular problem the system facilitates the identification of technicians with suitable experience for a specific situation. The prototype shows that relevant cases for the situation or problem can be retrieved whit a hybrid textual CBR approach. Also transferring experience through cases is a way that is favored by technicians and they recognize such a system as a useful tool giving them decision support in the form of a second opinion. This acceptance will also be validated in field tests in a real industrial environment.

References

Aamodt, A. and Plaza, E. (1994) “Case-based reasoning: foundational issues, methodological variations, and system approaches” AI Communications, 7 (1), pp. 39-59.

Blomberg A. (2005) “Matching Information According to Cognitive Preferences”, Masters Thesis Mälardalens University and KTH.

Delaître, S. and Moisan, S. (2000) “Knowledge Management by Reusing Experience”, 12th International Conference on Knowledge Engineering and Knowledge Management, pp. 304-311.

Dr. E. Garcia, “Tutorial on Cosine Similarity Measures”. http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html, Available 20-03-2006.

Funk, P. and Jackson, M. (2005) “Experience Based Diagnostics and Condition Based Maintenance within Production Systems”. In proceedings of the 18th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management (eds. Mba D. U., Rao R. B. K N.), pp. 105-111, Cranfield, UK.

Funk, P., Olsson, E., Bengtsson, M., Xiong, N. (2006) “Case-Based Experience Reuse and Agents for Efficient Health Monitoring, Preventive, and Corrective Actions”. In proceedings of the 19th International Congress on Condition Monitoring and Diagnostic Engineering Management, Luleå, Sweden.

Gruber T, “What is an Ontology?” http://www.ksl.stanford.edu/kst/what-is-an-ontology.html, Available 20-03-2006.

Marjanovic, O. (2005) ”Sharing and Reusing Learning Experiences — The Knowledge Management Perspective”, Advanced Learning Technologies, ICALT 2005, Fifth IEEE International Conference, pp. 707-709

Mladenic D. and M. Grobelnik., (1998) “Feature selection for classification based on text hierarchy”. Conf. Automated Learning and Discovery.

Recio J. A., Diaz-Agudo B., Gomez-Martin M. A., and Wiratunga N. (2005) “ExtendingjCOLIBRI for textual CBR”. In Procs. Of 6th International Conference on CBR, ICCBR 2005, volume 3620 of LNCS, pages 421-435. Springer -Verlang

Rosina, O., Weber, Kevin, D., Ashley, and Stefanie, B. (2005) "Textual case-based reasoning". The Knowledge Engineering Review, Vol. 00:0, 1–00., Cambridge University Press, Printed in the United Kingdom.

Salton G. and C. Buckley, (1987) "Term Weighting Approaches in Automatic Text Retrieval". Technical Report. UMI Order Number: TR87-881., Cornell University.

Salton G., A. Wong and C. S. Yang. (1975) "A Vector Space Model for Automatic Indexing". Communications of the ACM, vol.18, nr. 11, pages 613–620.

3

Scott S. and S. Matwin. (1998) "Text Classification Using WordNet Hypernyms", Use of WordNet in Natural Language Processing Systems.

Swartout W. (1999) "Ontologies", IEEE Intelligent Syst., (Jan./Feb.), 18–19.

Watson. (1999) "Case-based reasoning is a methodology not a technology", AI-CBR, University of Salford, Salford M5 4WT, UK.

Watson, I. (1997) "Applying Case-Based Reasoning: Techniques for Enterprise Systems", Morgan Kaufmann Publishers Inc, 340 Pine St, 6th floor, San Fransisco, CA 94104, USA.

Wilson, D. C. and Bradshaw, S. (2000) "CBR Textuality", Expert Update, 3(1). pp. 28-37.